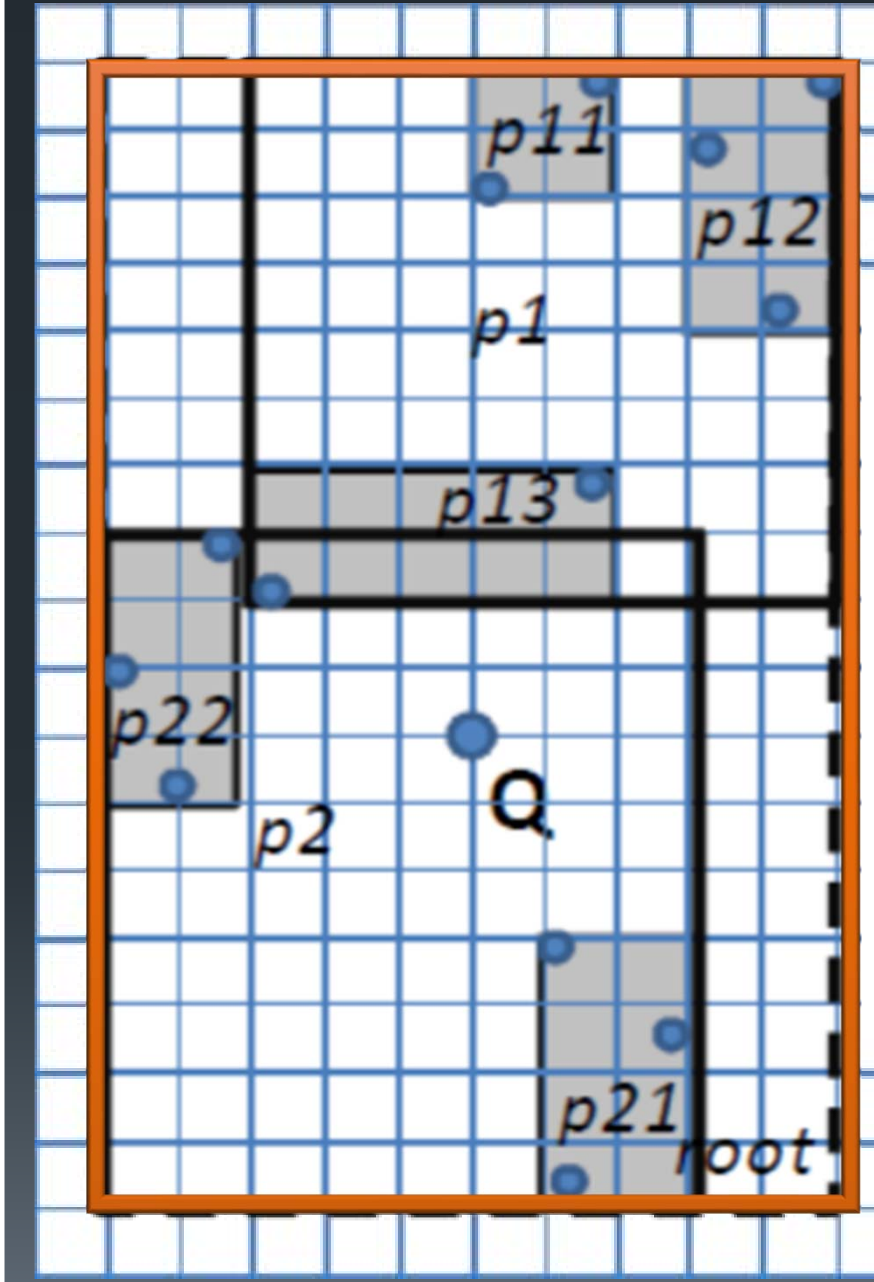
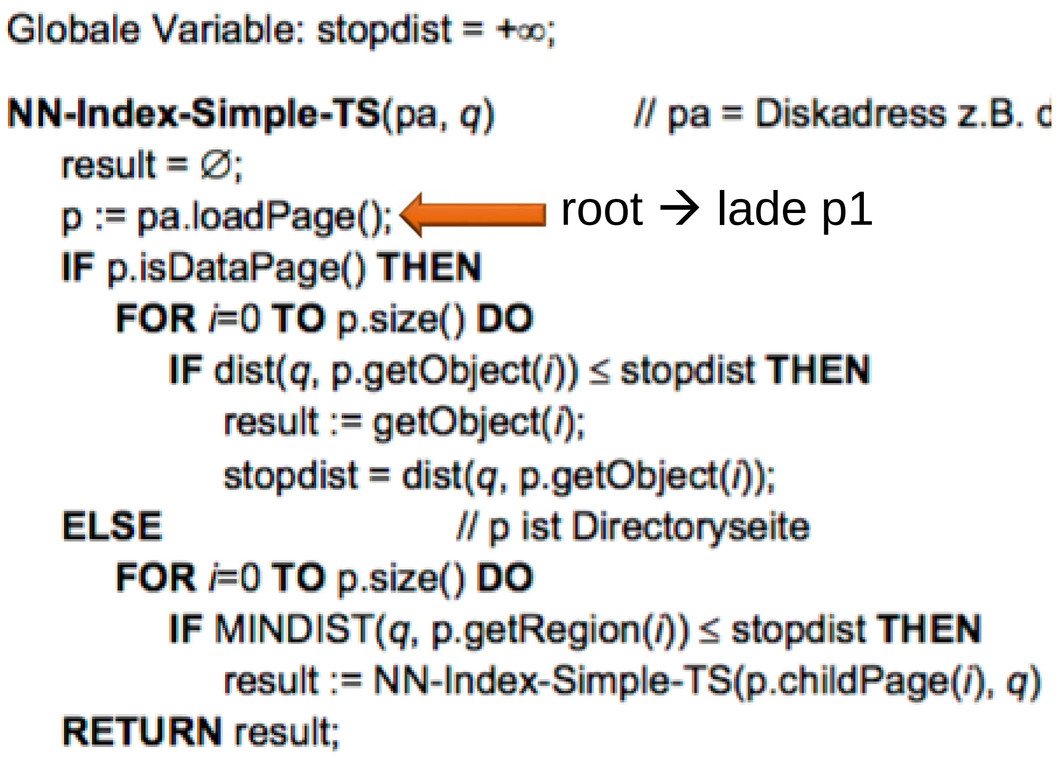
STMD – ÜB5

A5-2 (a)

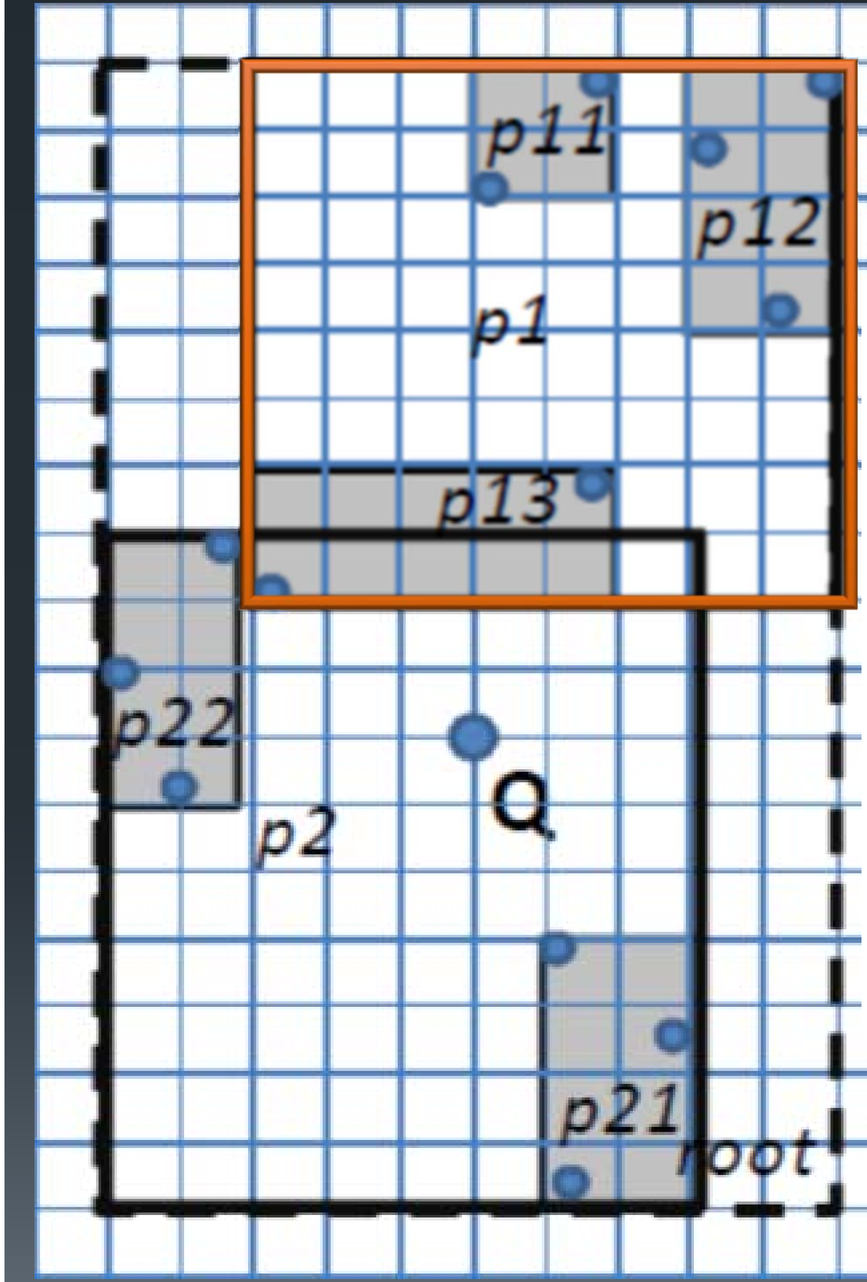


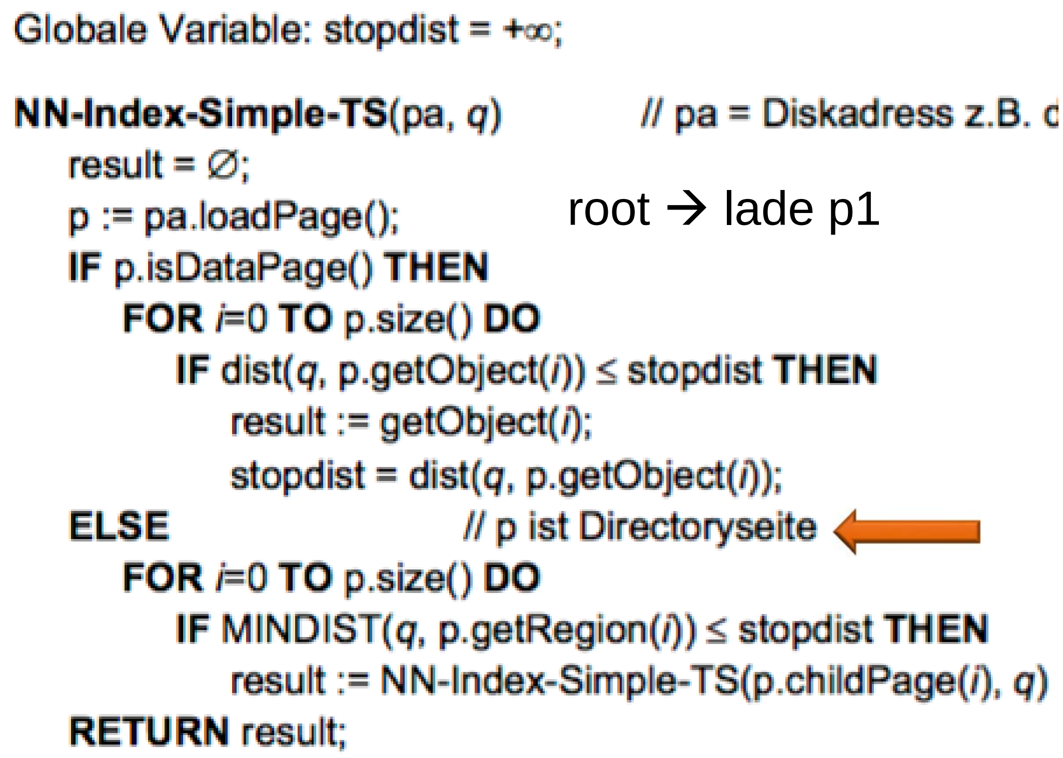


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

root → lade p1

Besuchte Seiten: p1

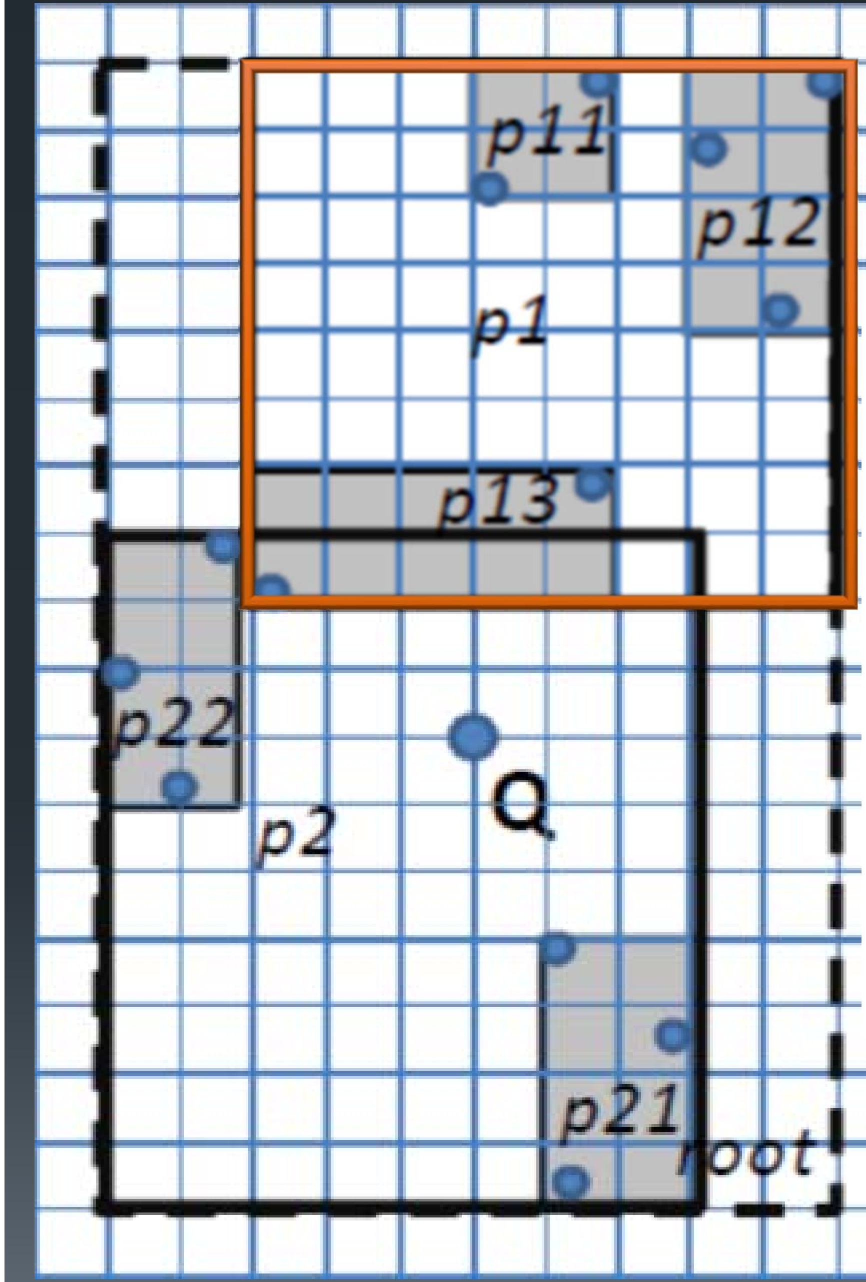


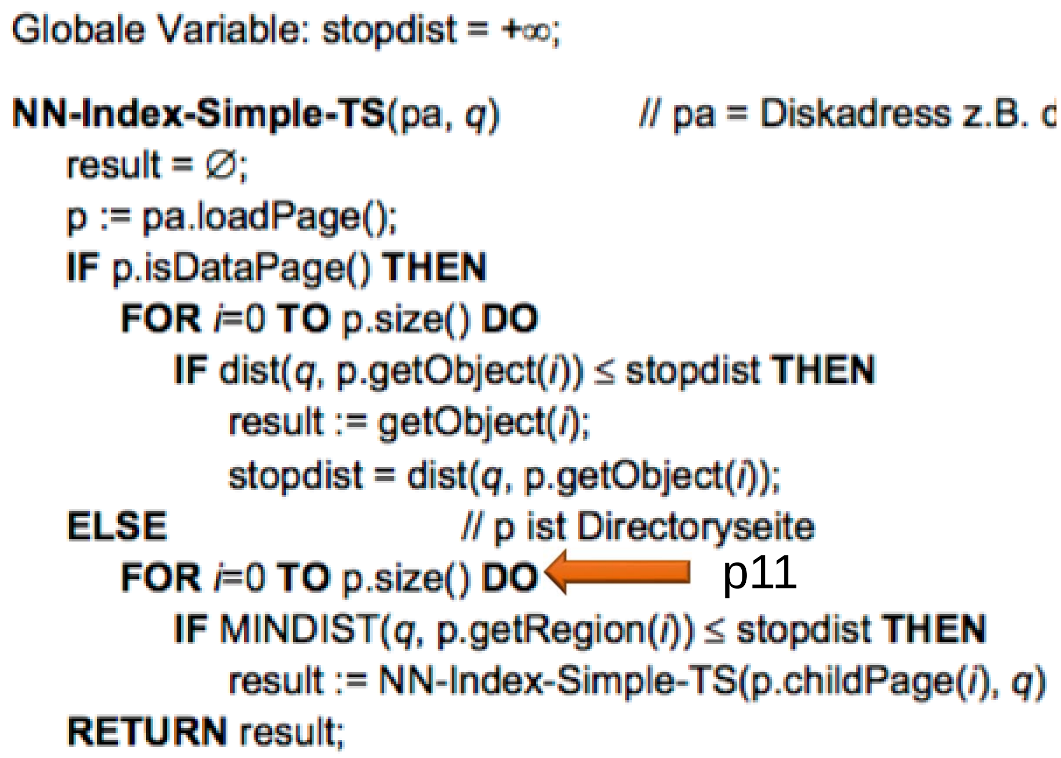


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                    // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

root → lade p1

Besuchte Seiten: p1

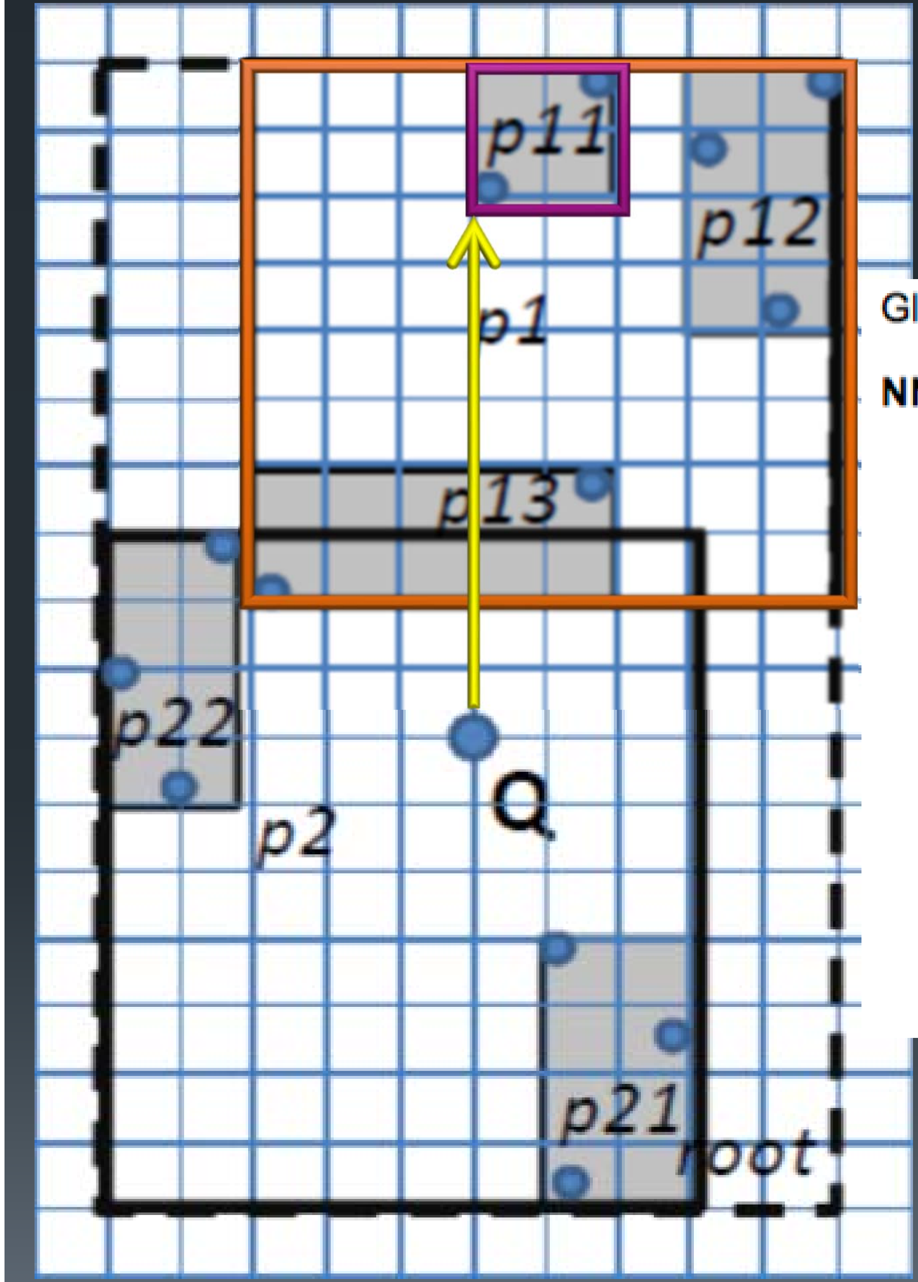


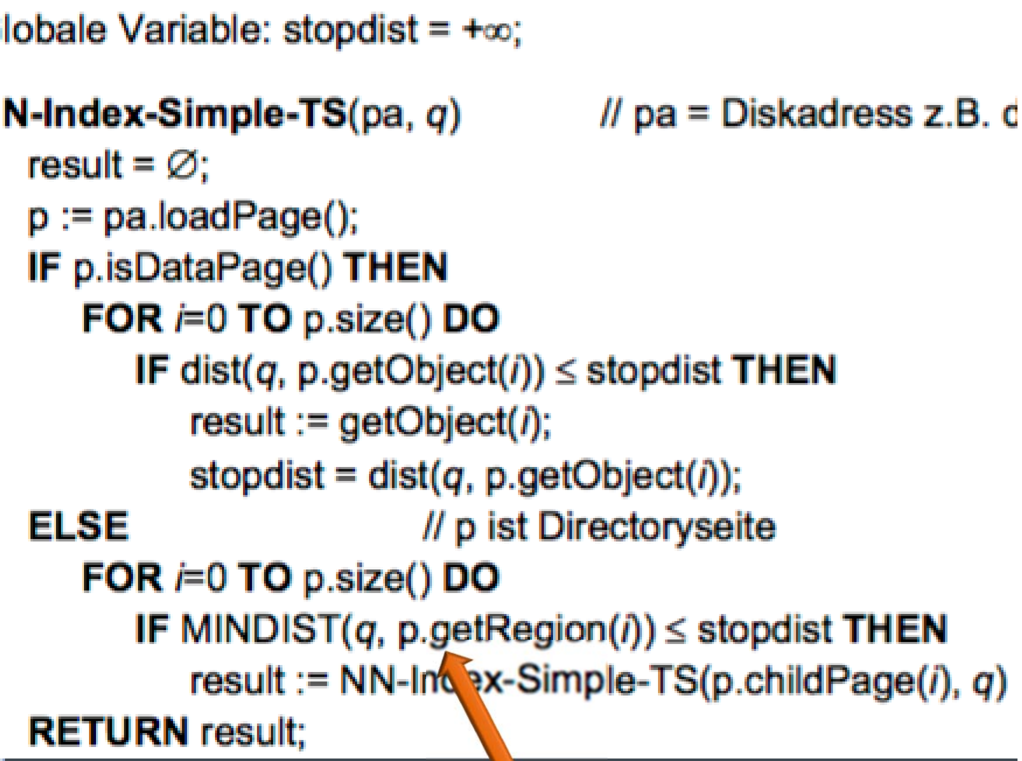


Globale Variable: stopdist = +∞;

```
NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO ← p11
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Besuchte Seiten: p1



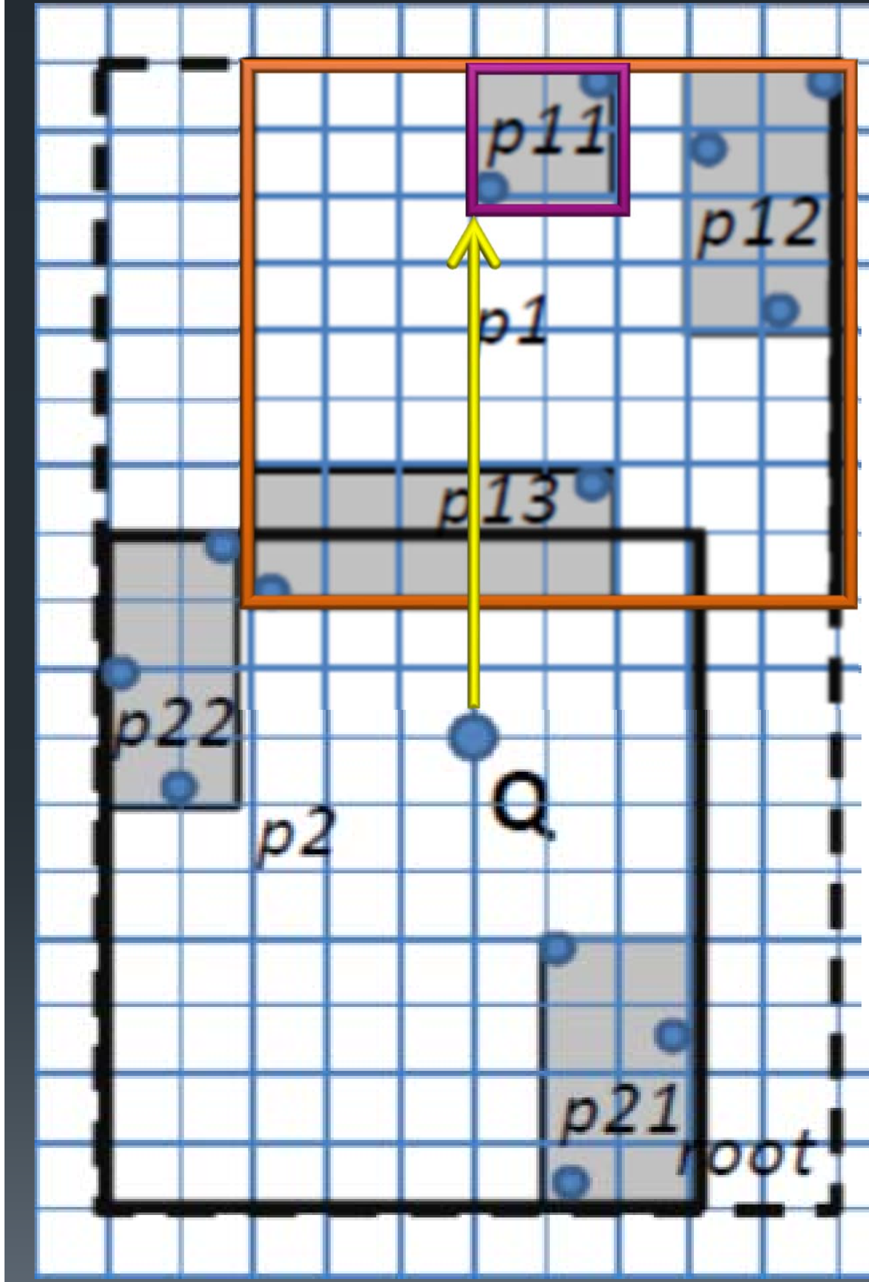


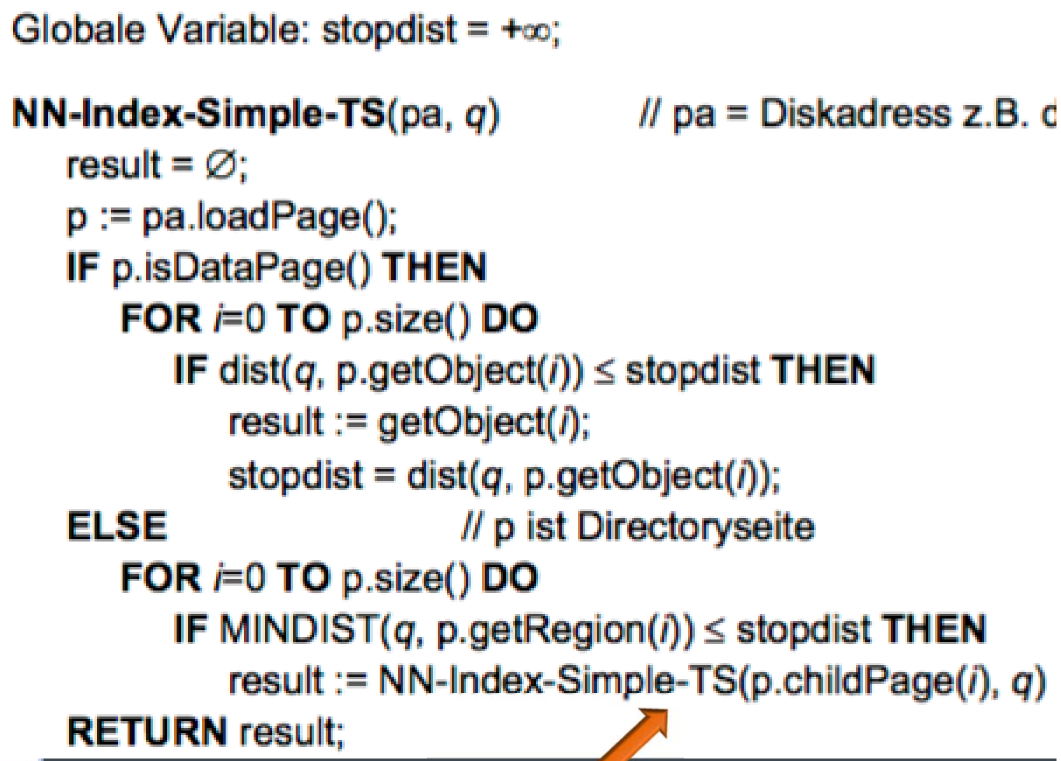
Globale Variable: stopdist = +∞;

```
NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

MINDIST(Q,p11)=8 <= Inf? → ja!

Besuchte Seiten: p1,p11

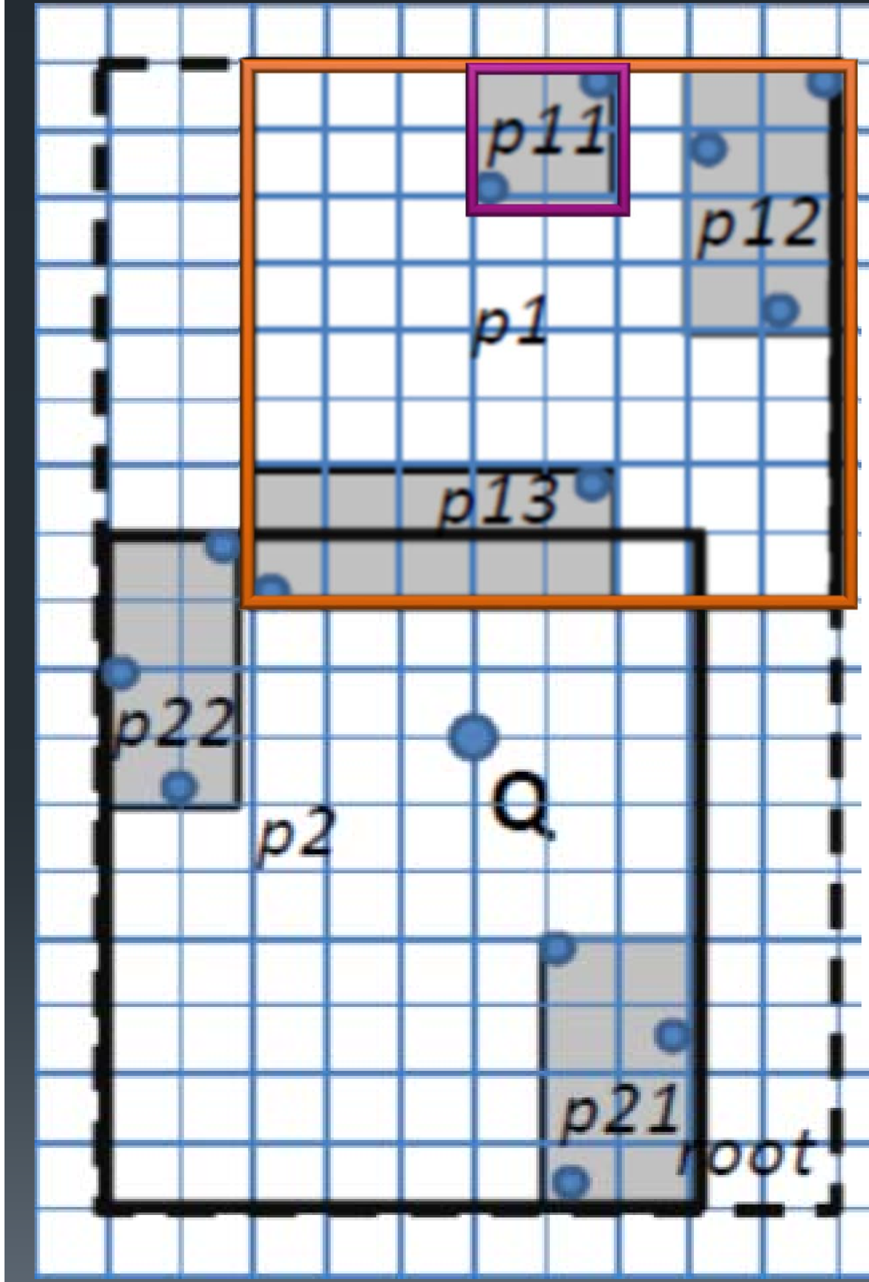


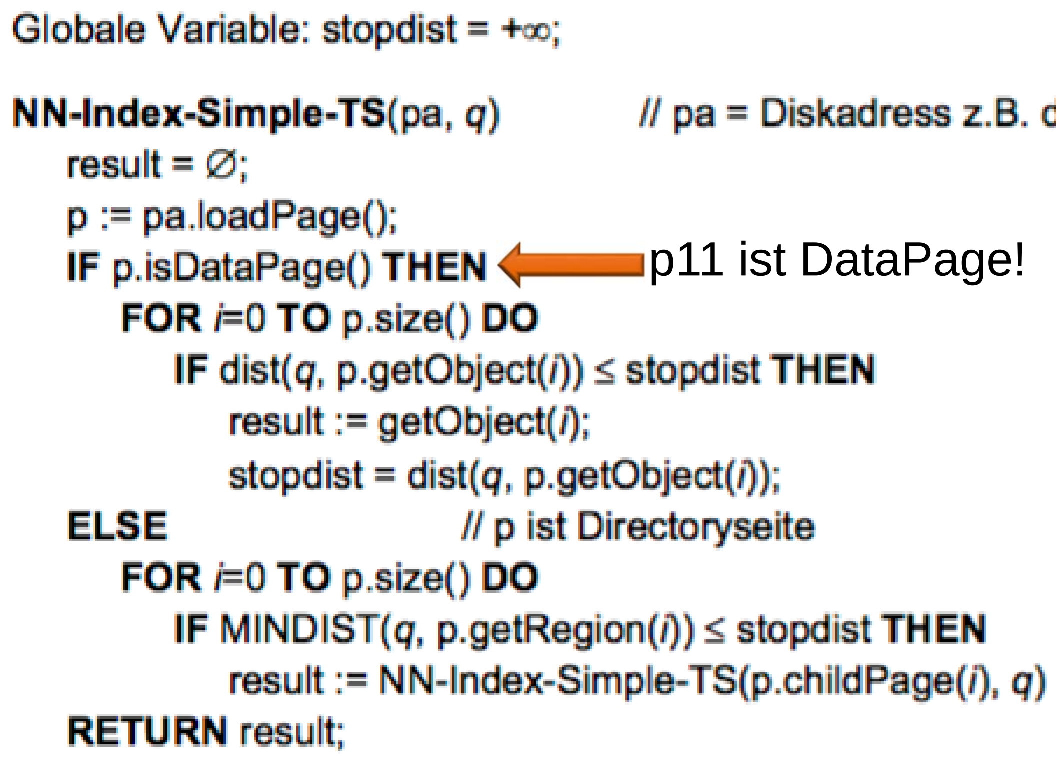


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                         // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

rekursiver Aufruf

Besuchte Seiten: p1,p11

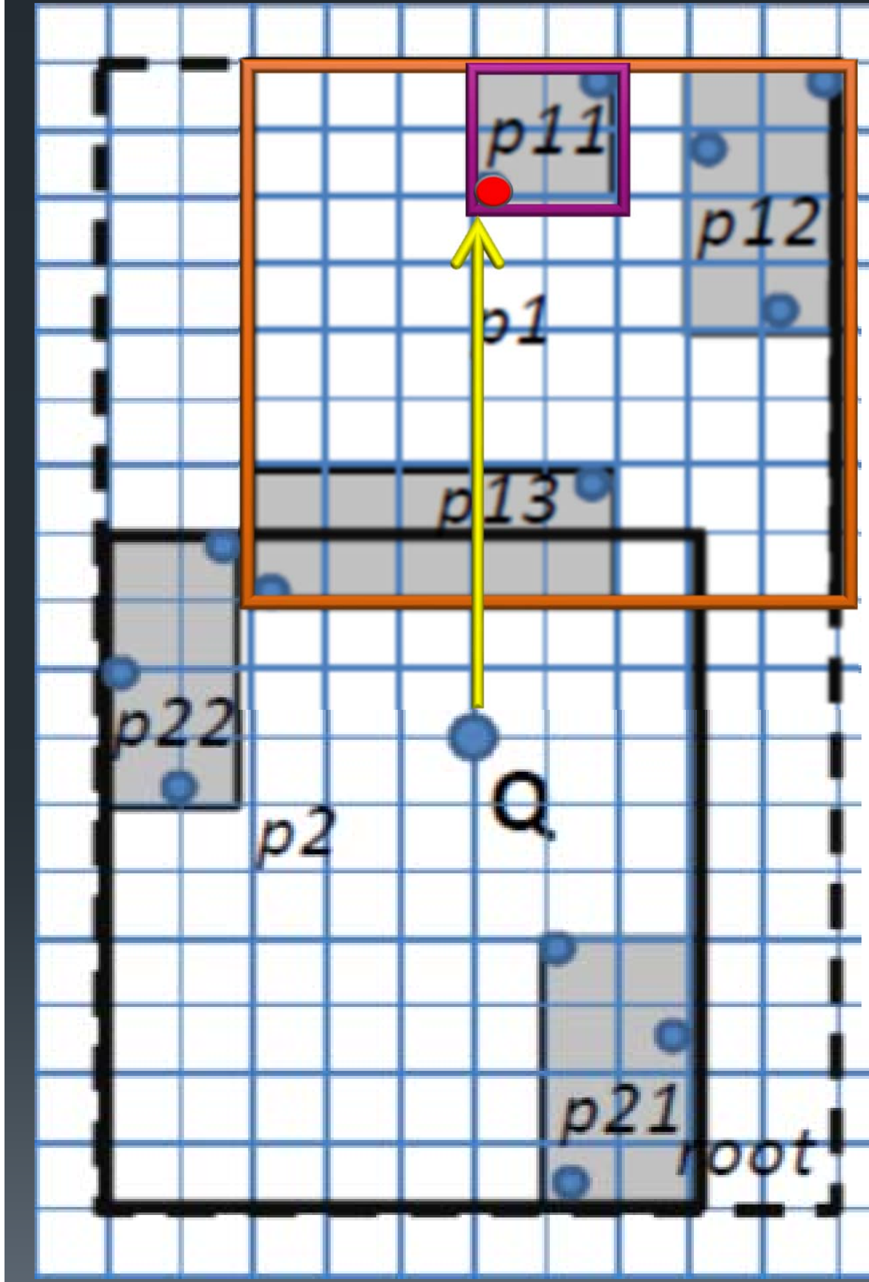


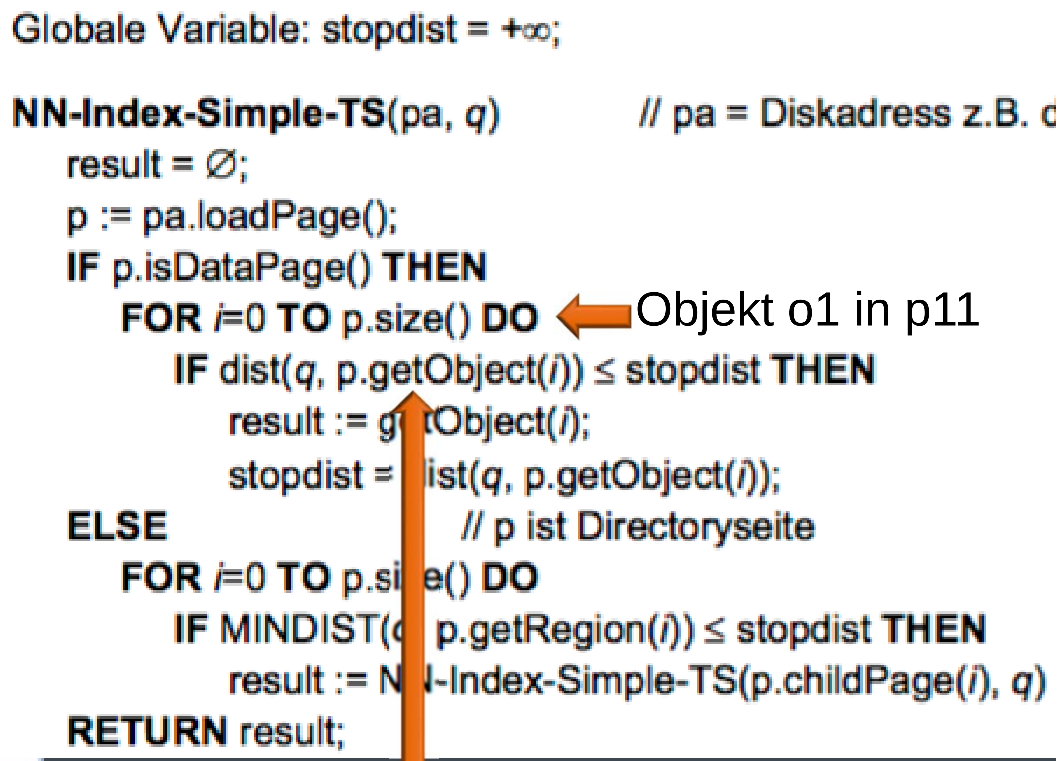


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                              // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

p11 ist DataPage!

Besuchte Seiten: p1,p11



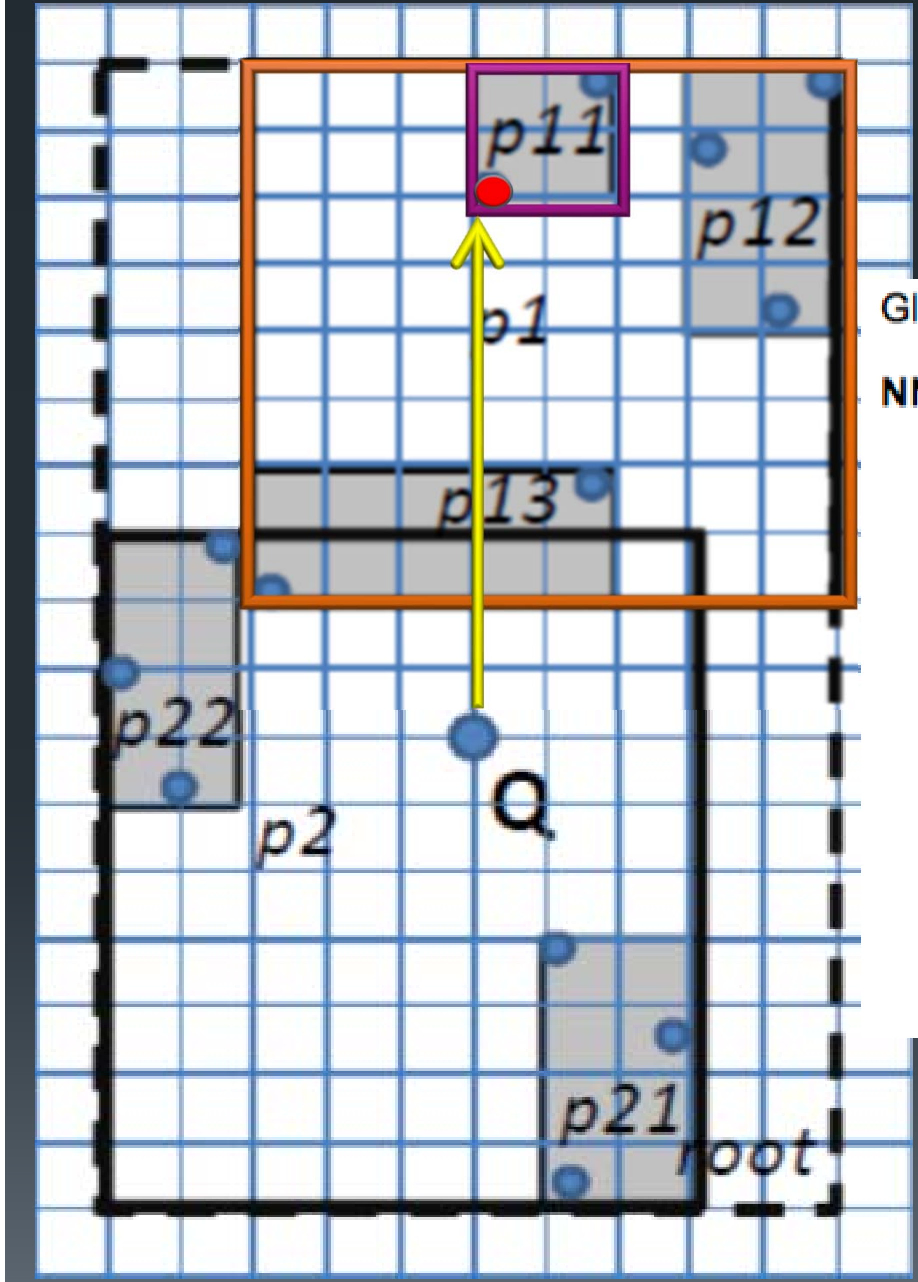


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Objekt o1 in p11

dist(Q,o1) = 8 <= Inf

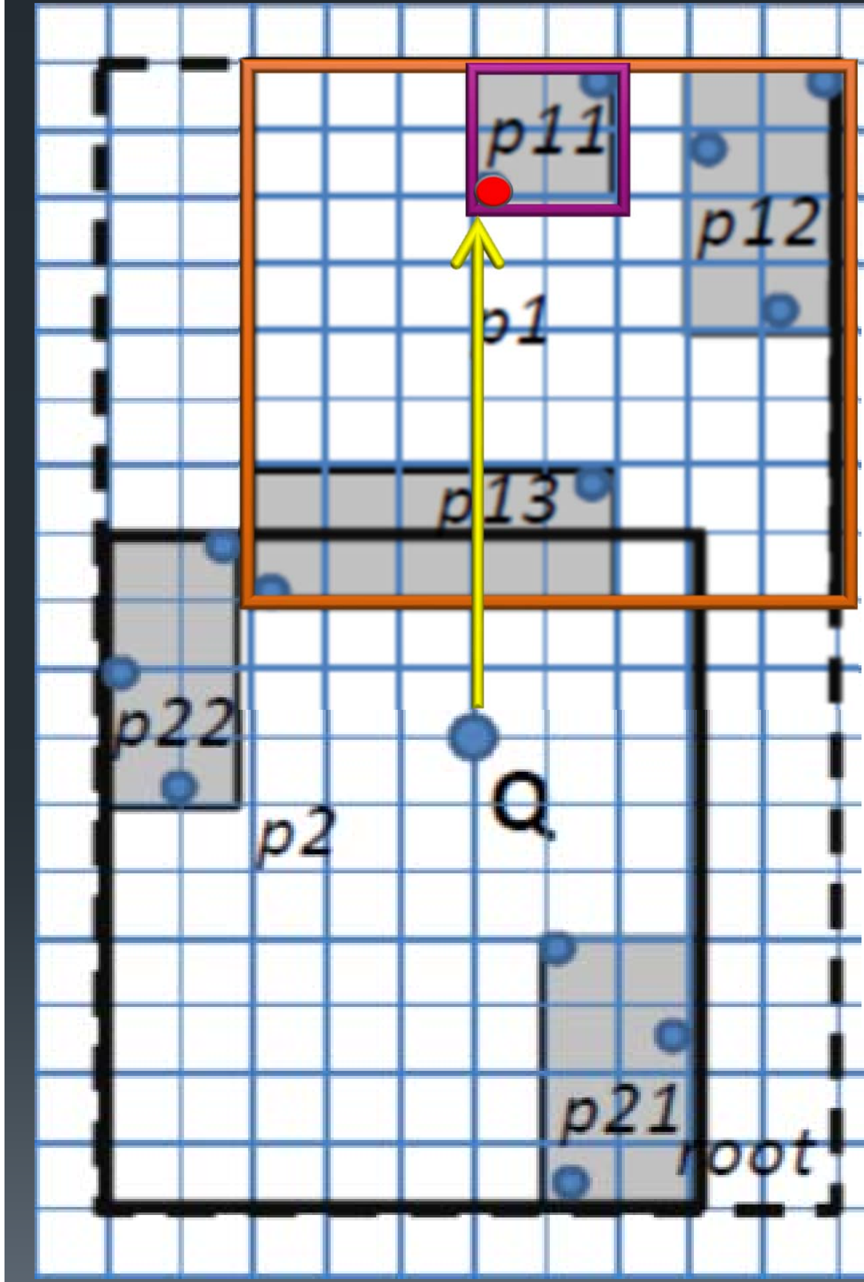
Besuchte Seiten: p1,p11



Globale Variable: stopdist = +∞;

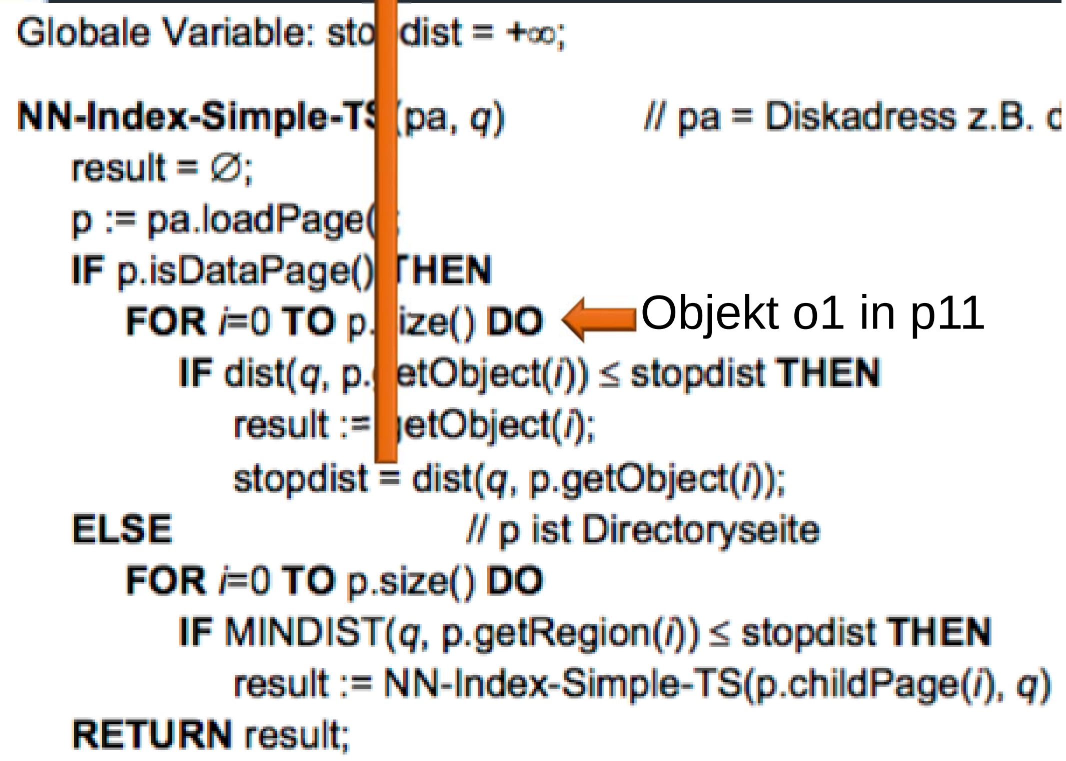
```
NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO ← Objekt o1 in p11
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i); ← o1 ist true hit
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Besuchte Seiten: p1,p11



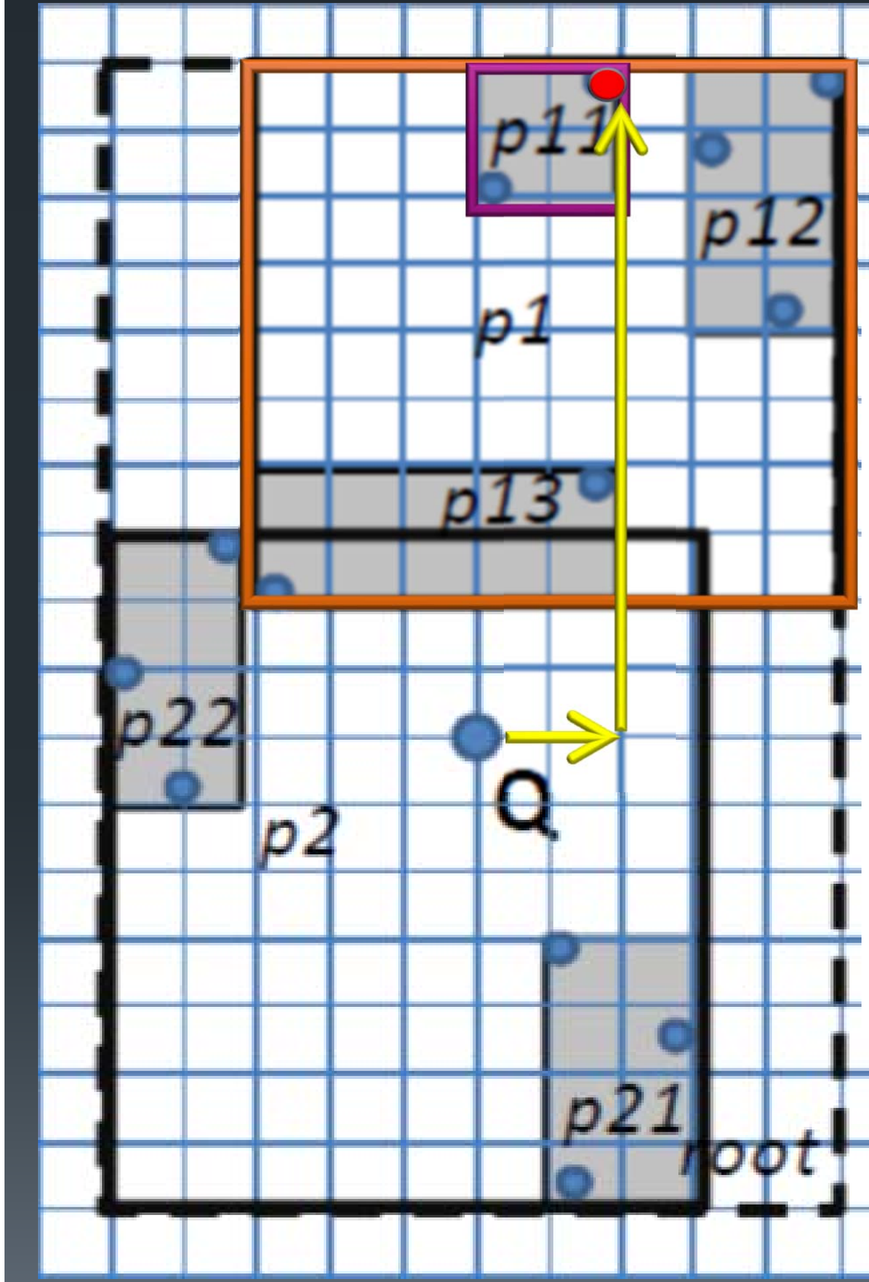
stopdist = dist(Q,o1) = 8

Objekt o1 in p11

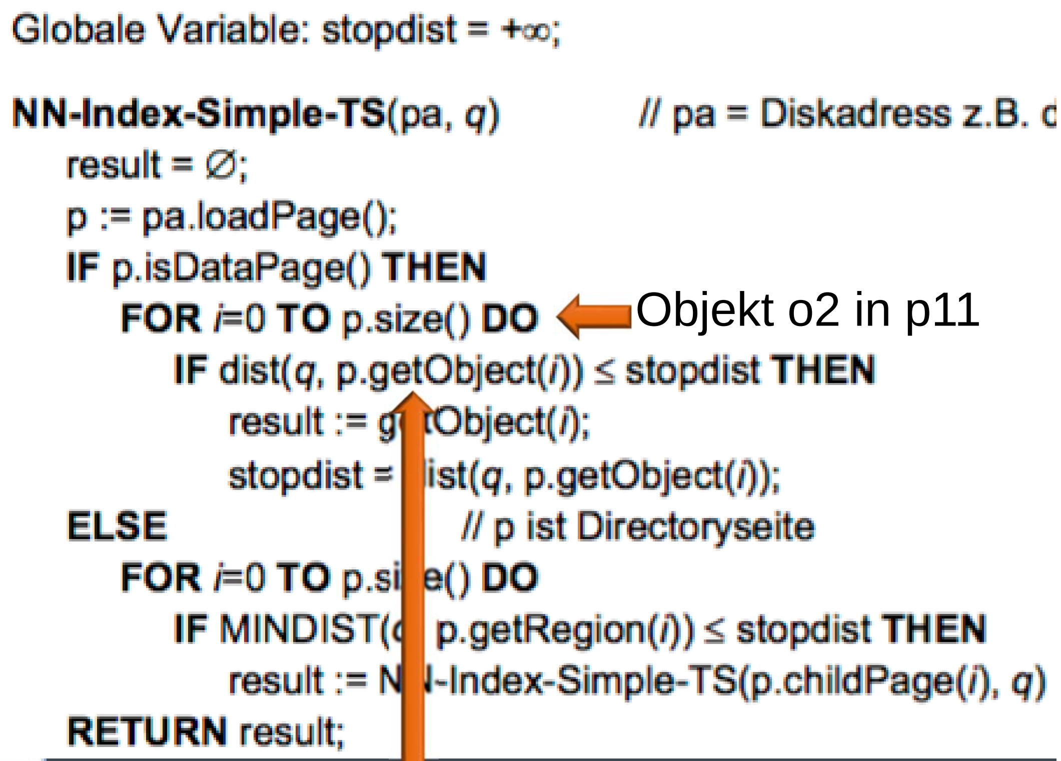


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Besuchte Seiten: p1,p11



stopdist = dist(Q,o1) = 8



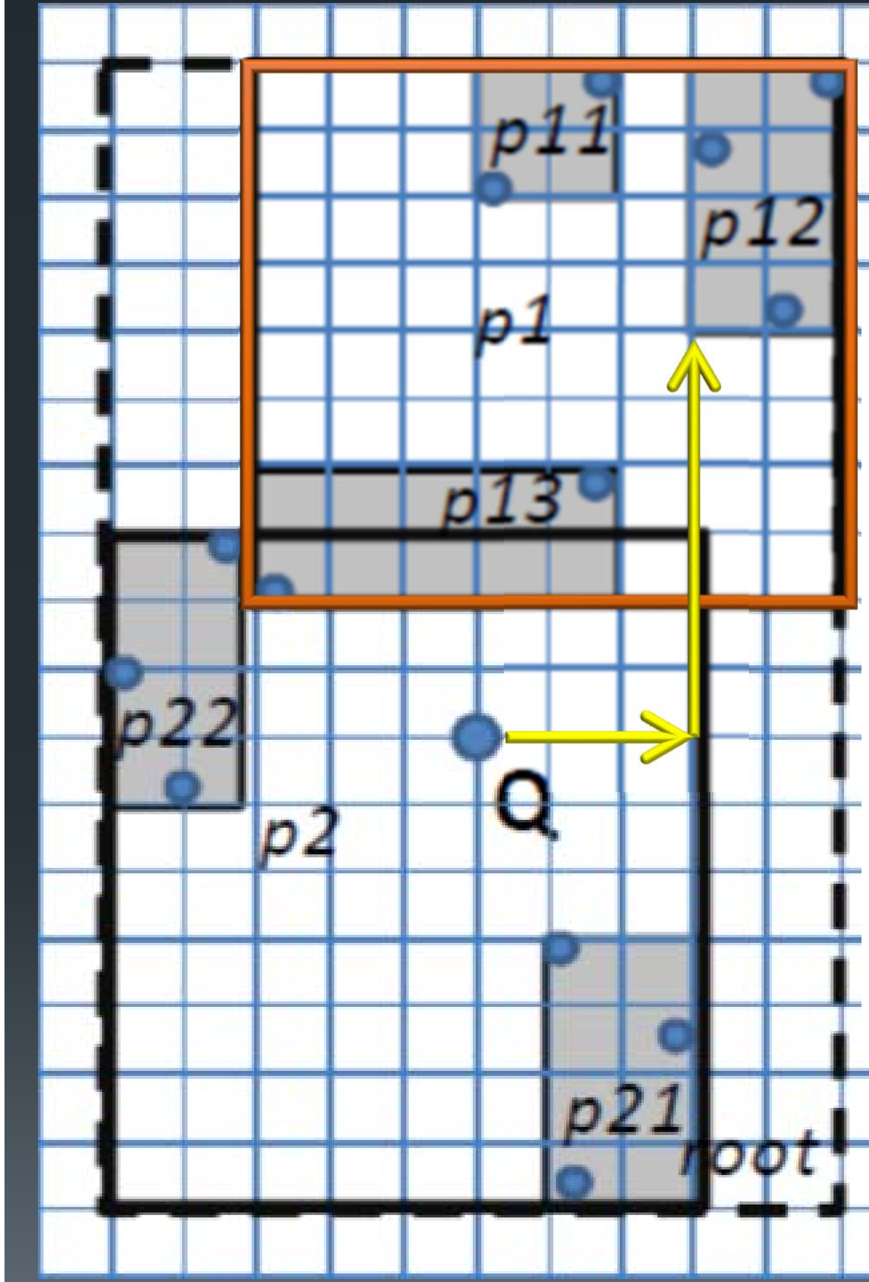
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
  result = ∅;
  p := pa.loadPage();
  IF p.isDataPage() THEN
    FOR i=0 TO p.size() DO
      IF dist(q, p.getObject(i)) ≤ stopdist THEN
        result := getObject(i);
        stopdist = dist(q, p.getObject(i));
  ELSE                               // p ist Directoryseite
    FOR i=0 TO p.size() DO
      IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
        result := NN-Index-Simple-TS(p.childPage(i), q)
  RETURN result;
```

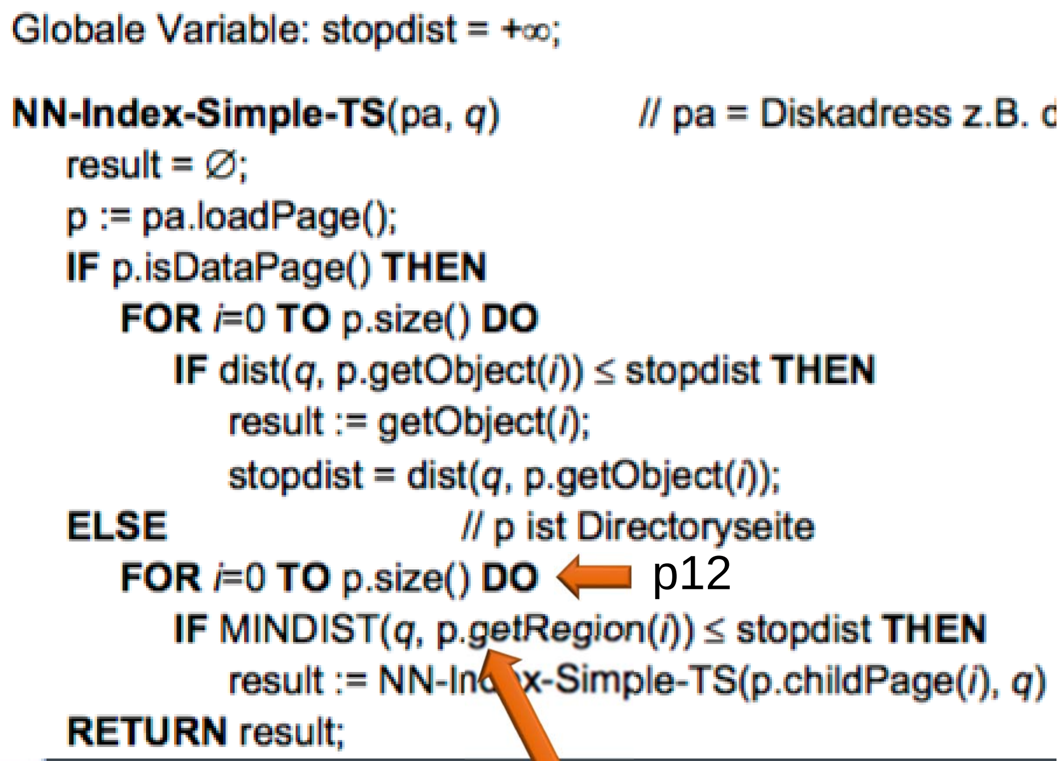
Objekt o2 in p11

dist(Q,o2) = 12 <= 8?
Nein → true drop

Besuchte Seiten: p1,p11

stopdist = dist(Q,o1) = 8



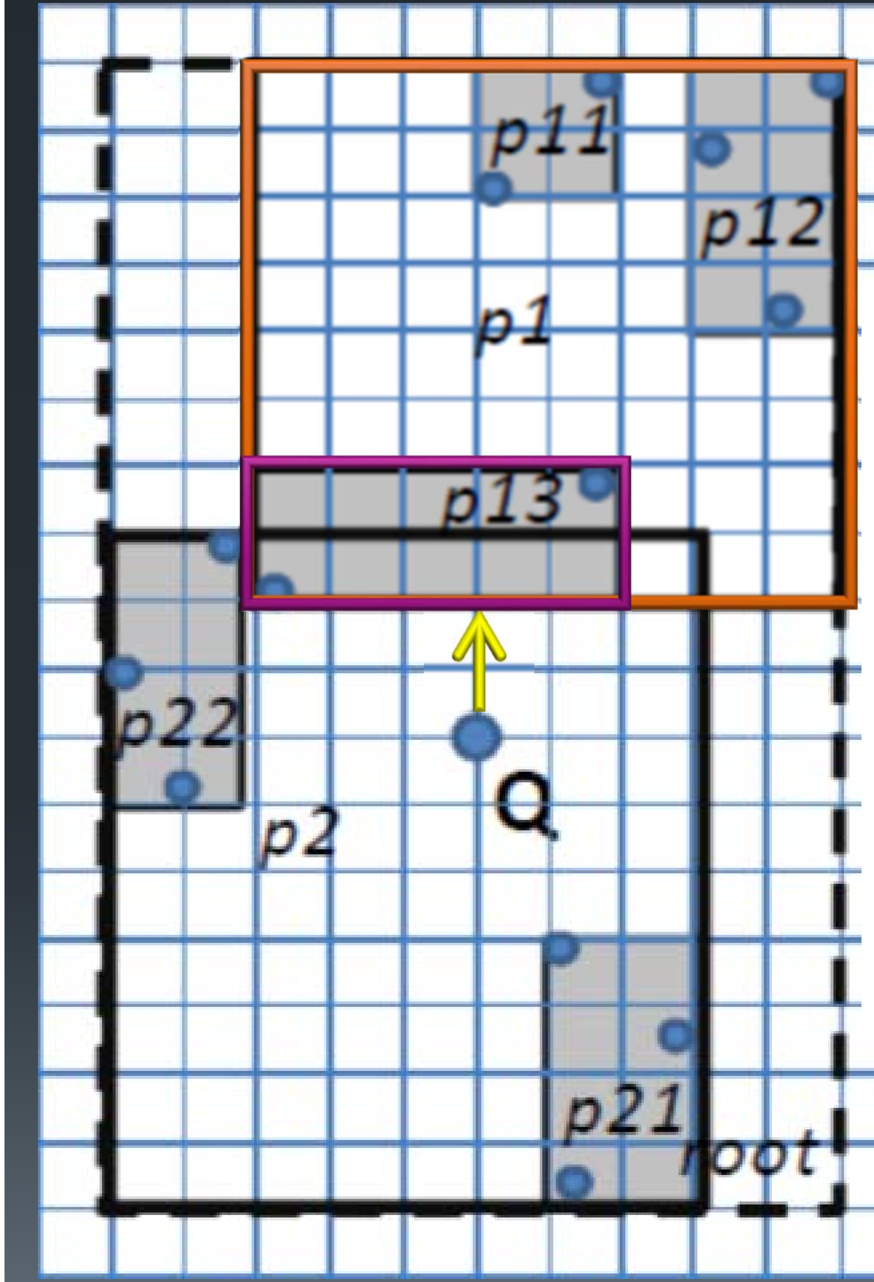


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                        // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

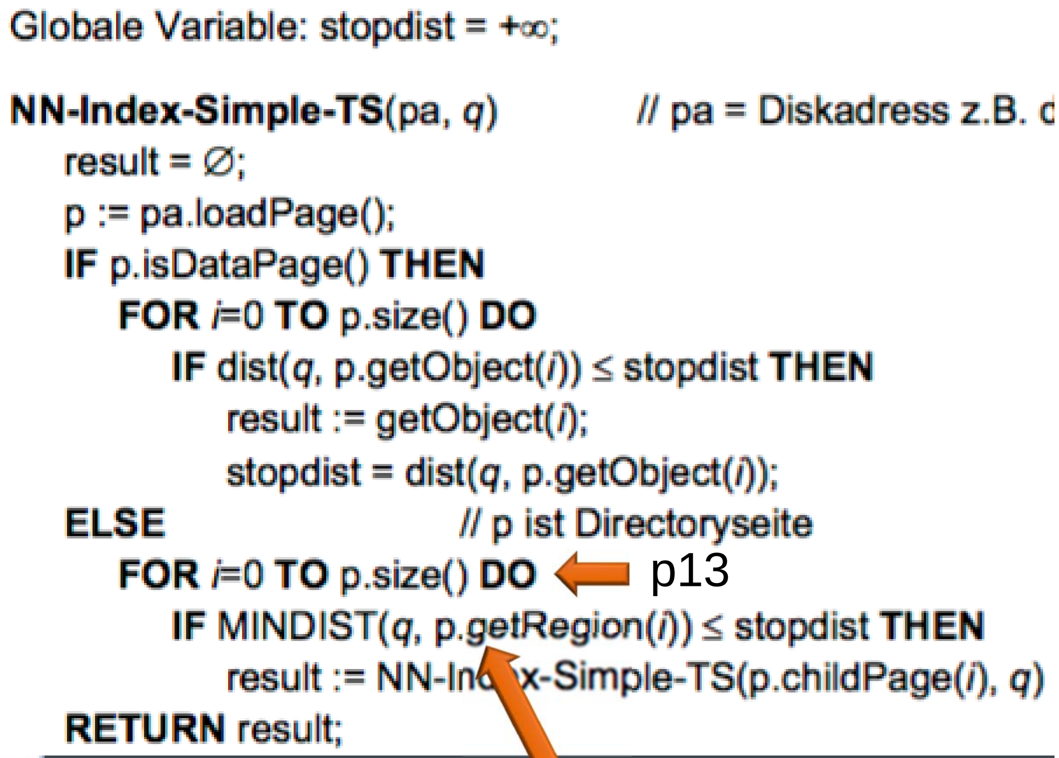
p12

MINDIST(Q,p12)=9 <= 8?
Nein! Verwirf p12!

Besuchte Seiten: p1,p11



stopdist = dist(Q,o1) = 8

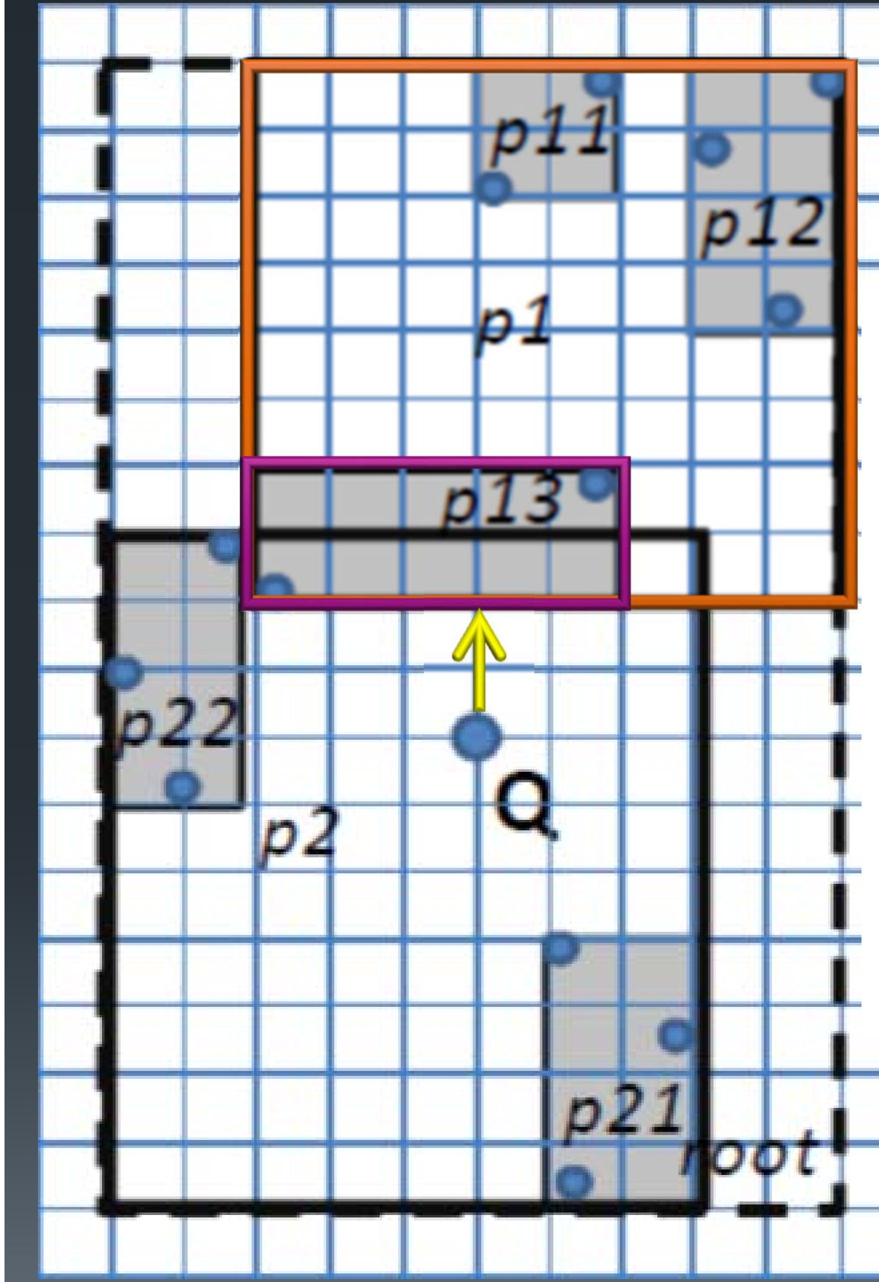


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

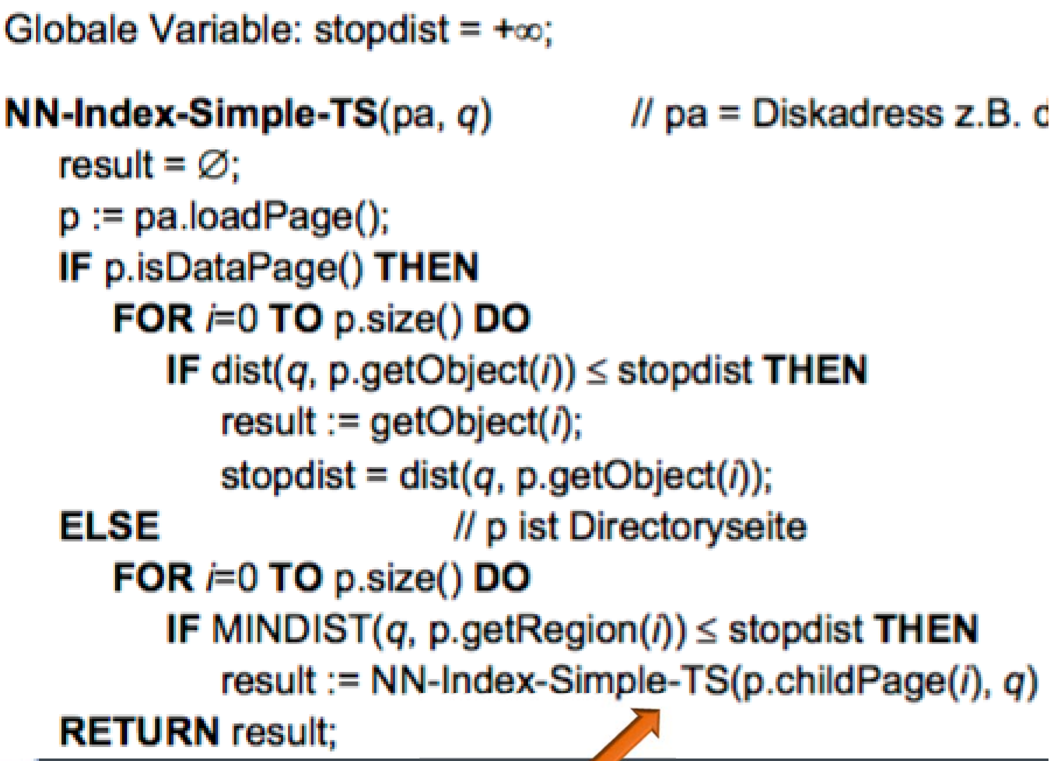
p13

MINDIST(Q,p13)=2 <= 8? → ja!

Besuchte Seiten: p1,p11,p13



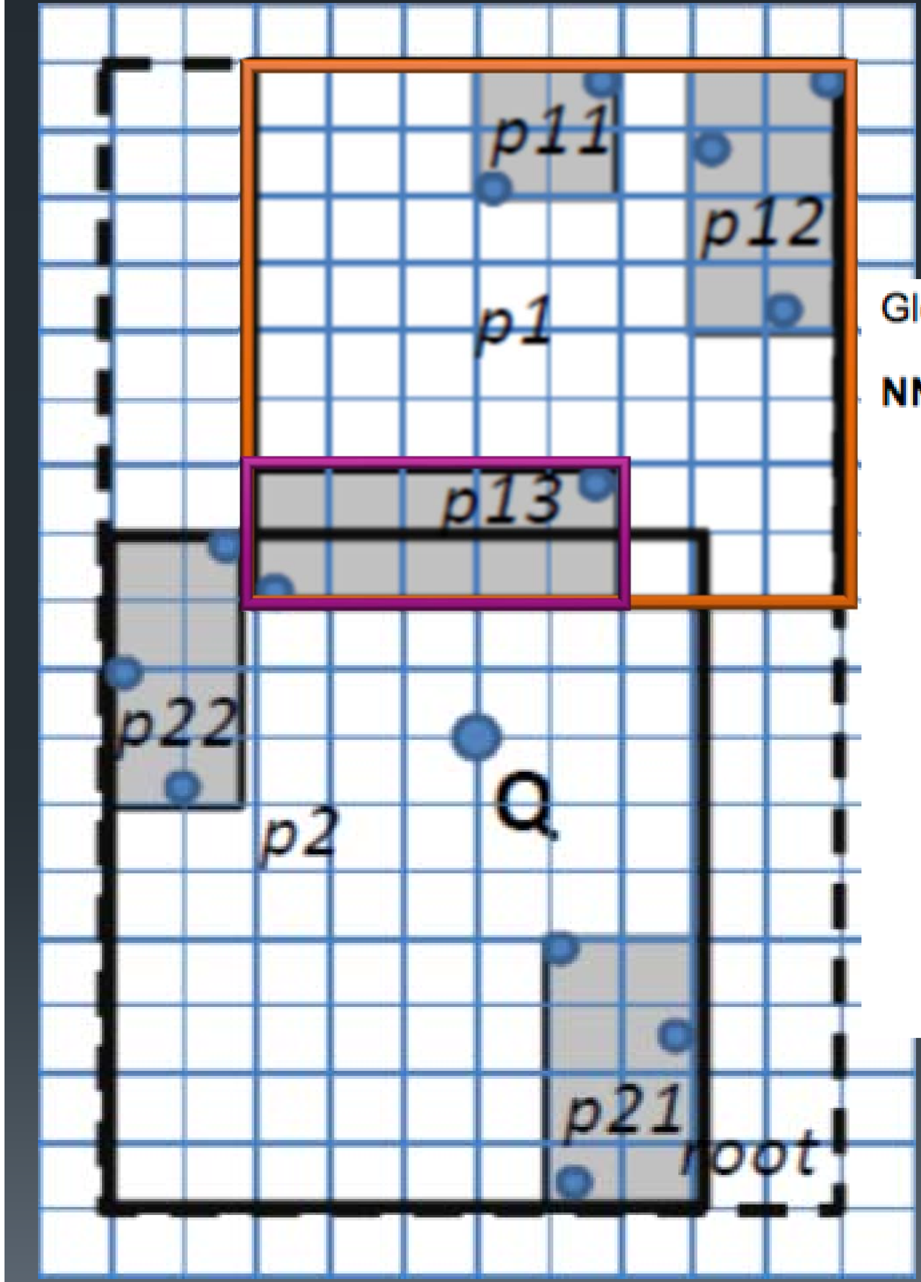
stopdist = dist(Q,o1) = 8



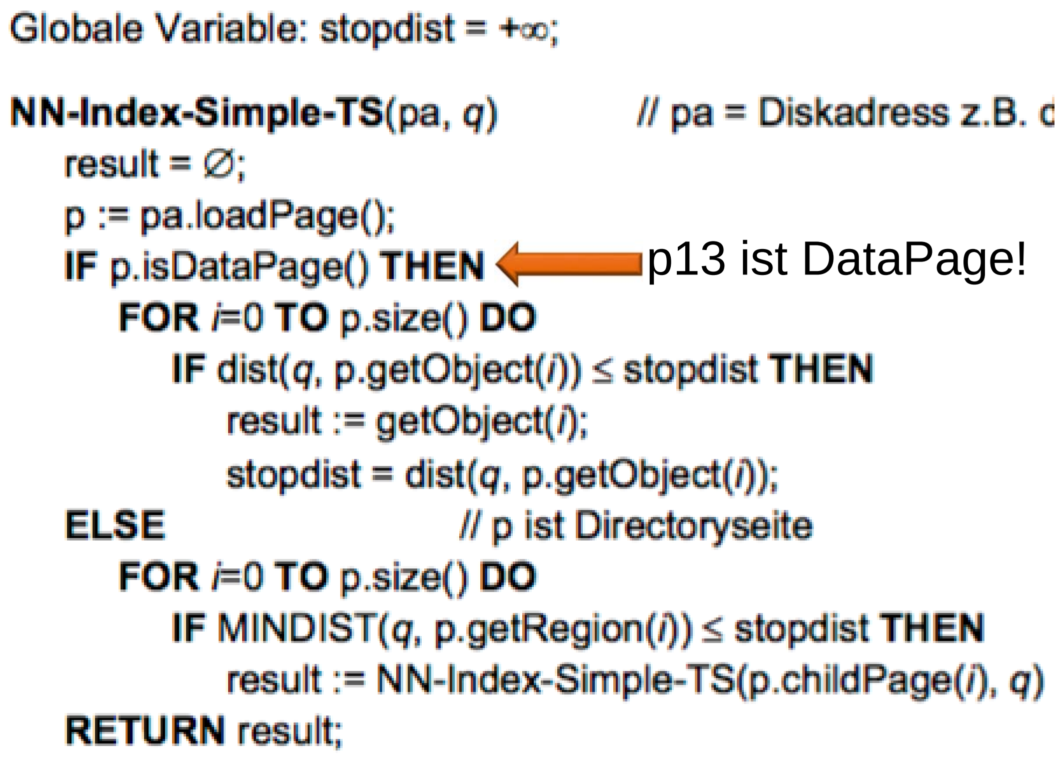
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                         // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

rekursiver Aufruf

Besuchte Seiten: p1,p11,p13



stopdist = dist(Q,o1) = 8

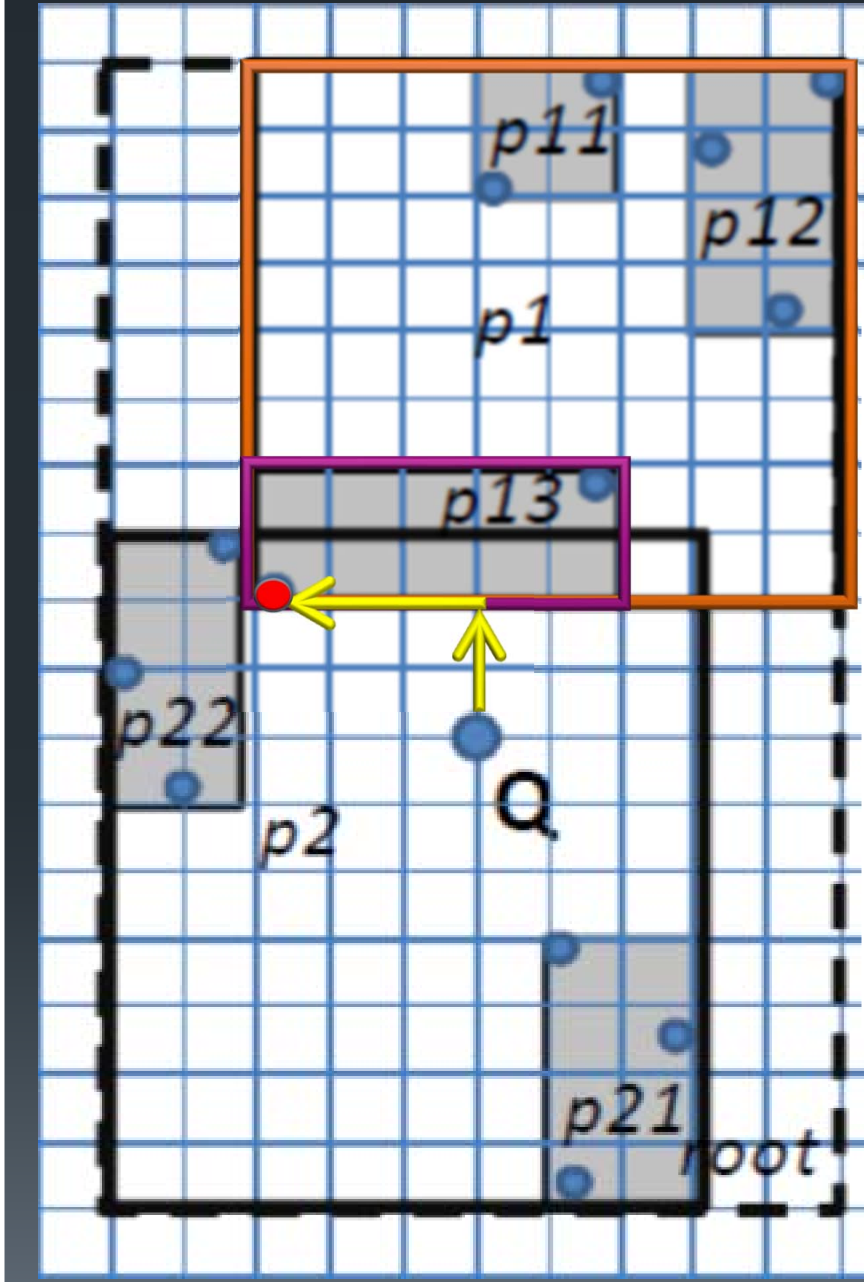


Globale Variable: stopdist = +∞;

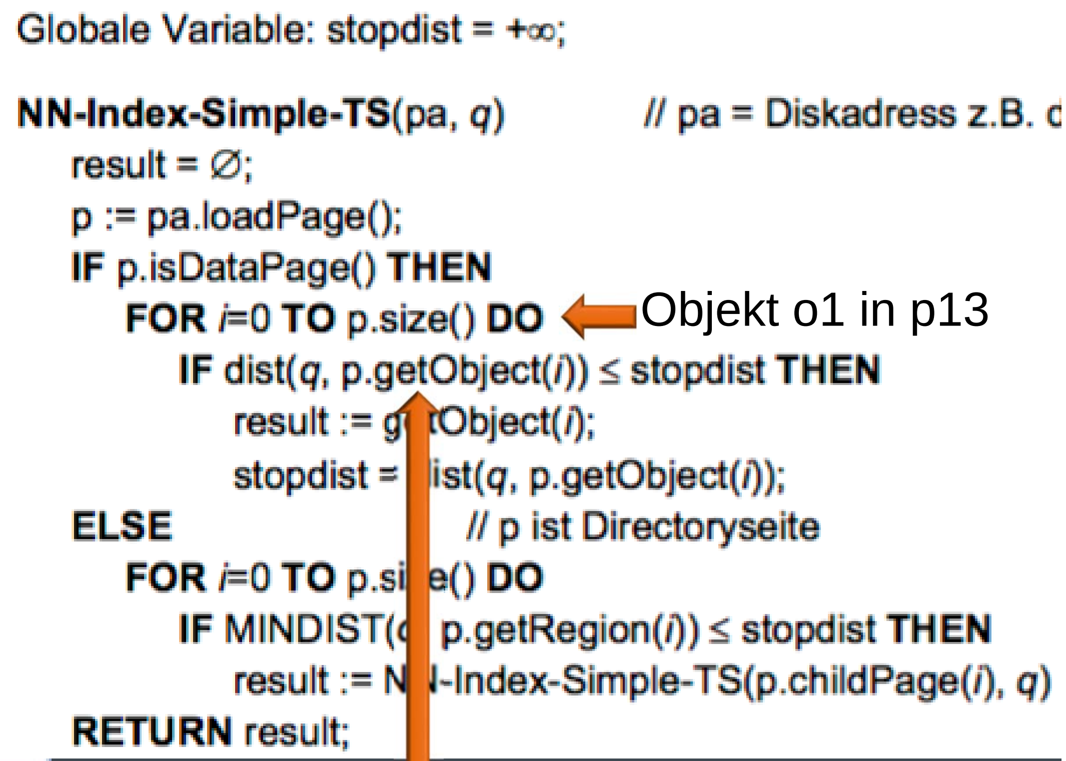
```
NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                            // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

p13 ist DataPage!

Besuchte Seiten: p1,p11,p13



stopdist = dist(Q,o1) = 8

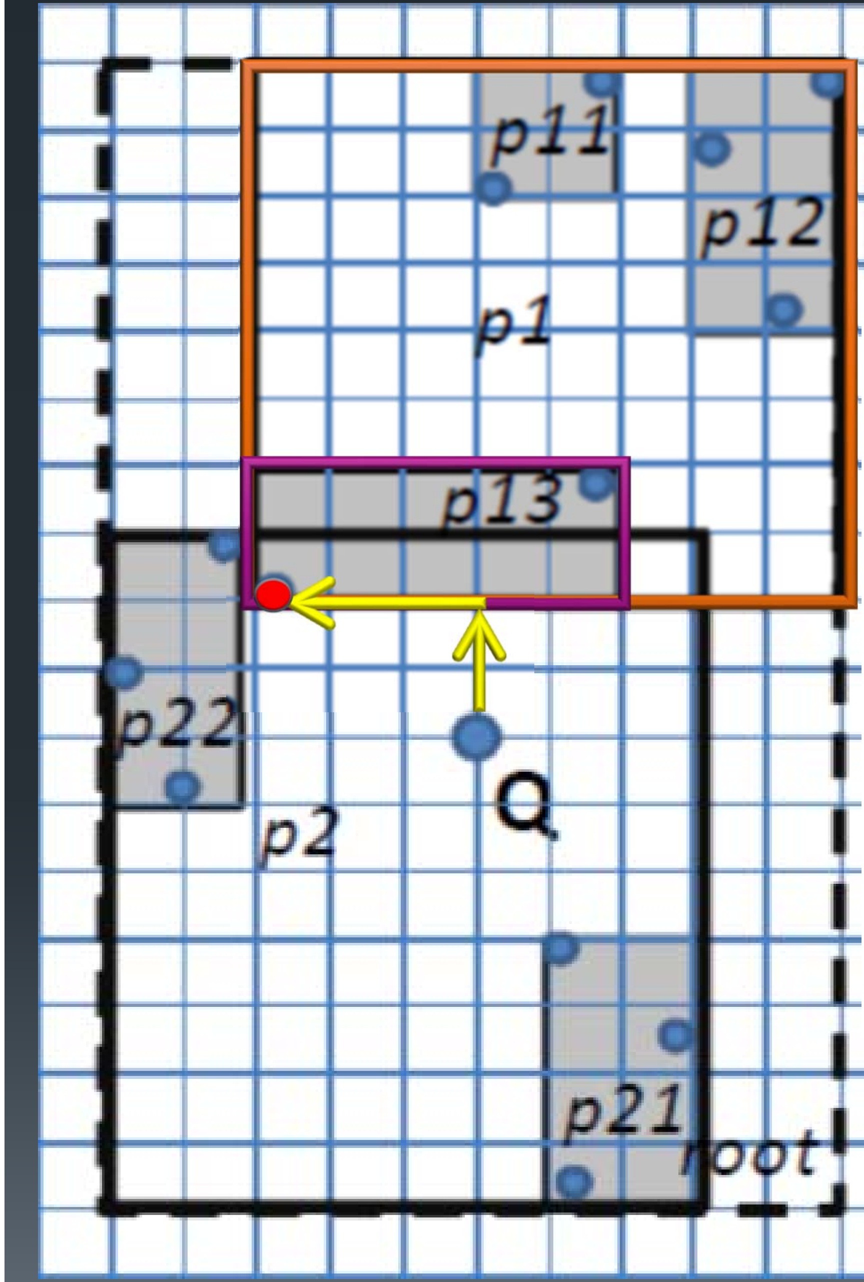


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

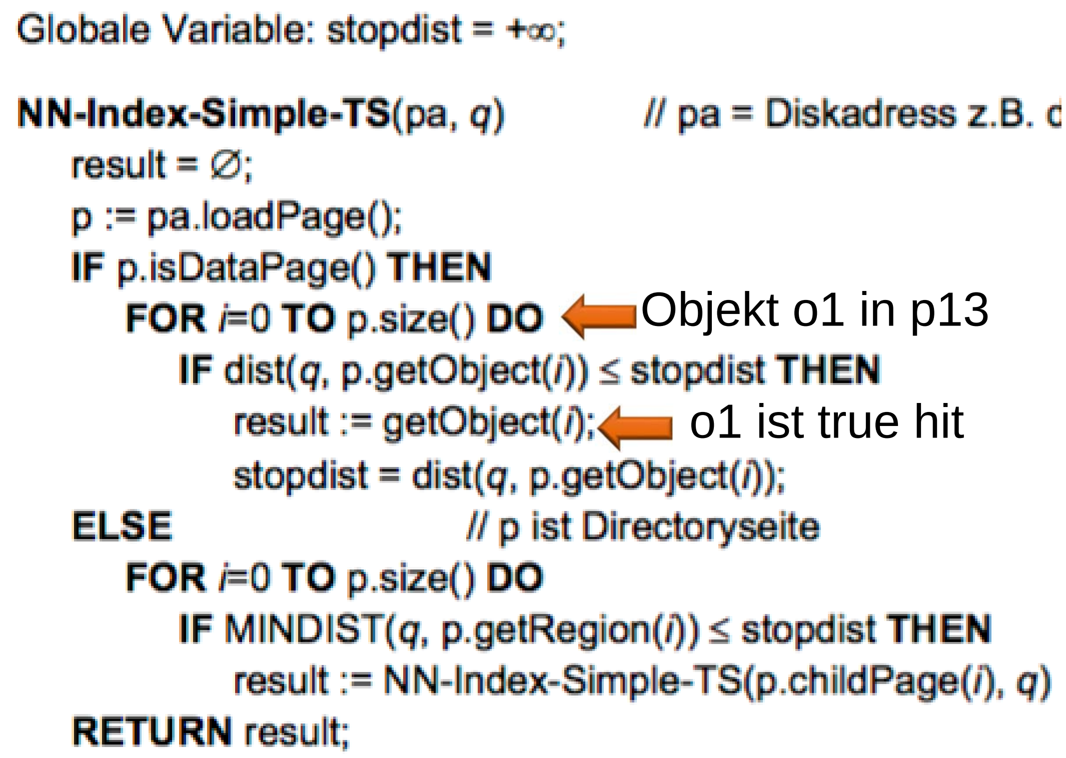
Objekt o1 in p13

dist(Q,o1) = 5 <= 8

Besuchte Seiten: p1,p11,p13



stopdist = dist(Q,o1) = 8

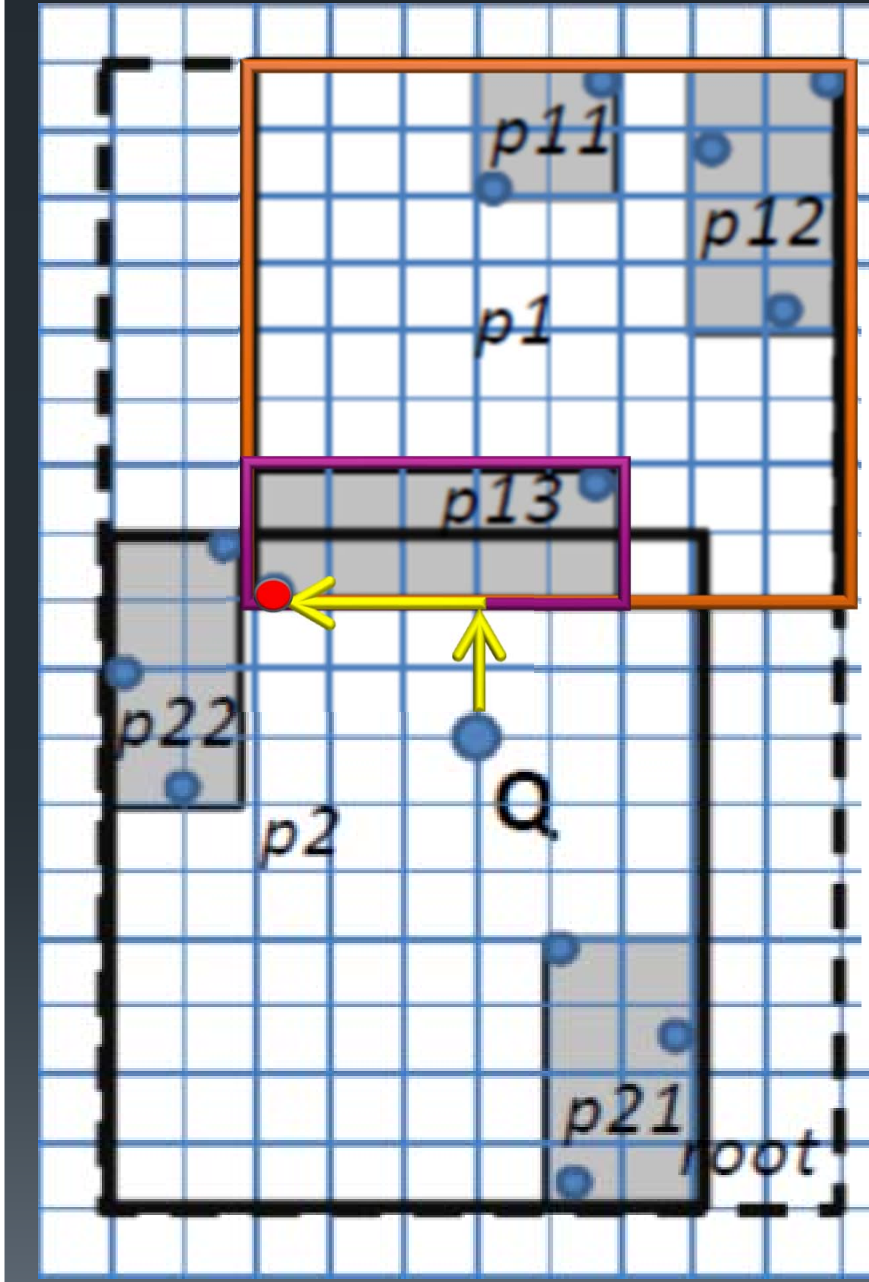


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

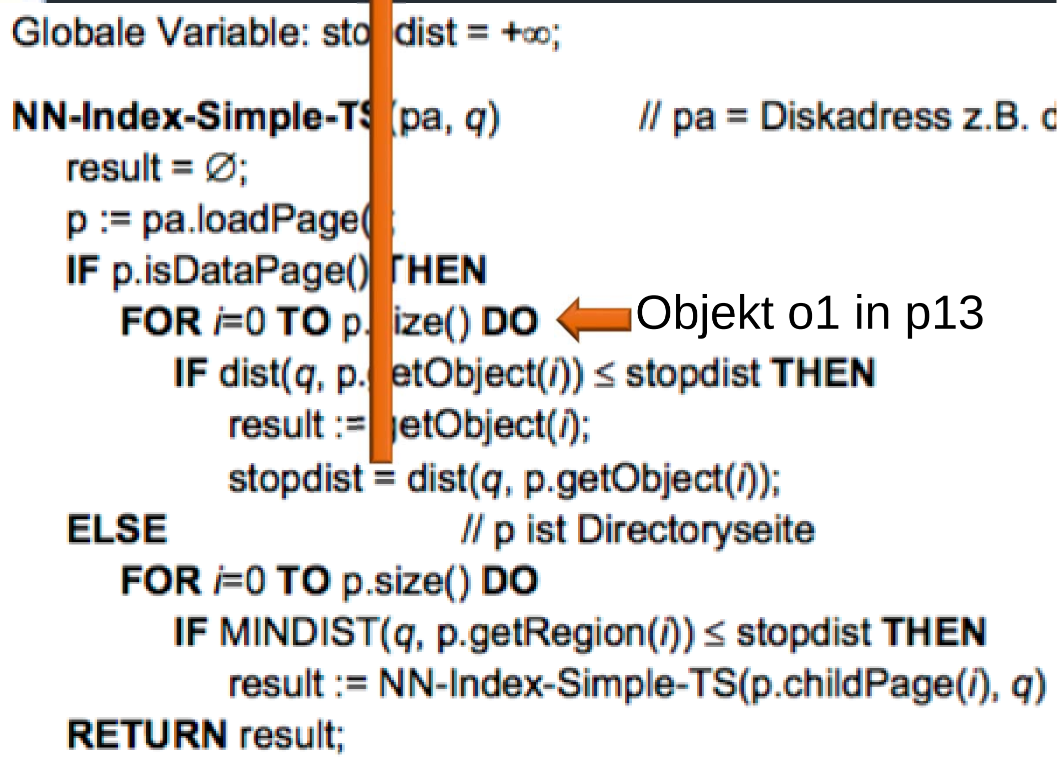
Objekt o1 in p13

o1 ist true hit

Besuchte Seiten: p1,p11,p13



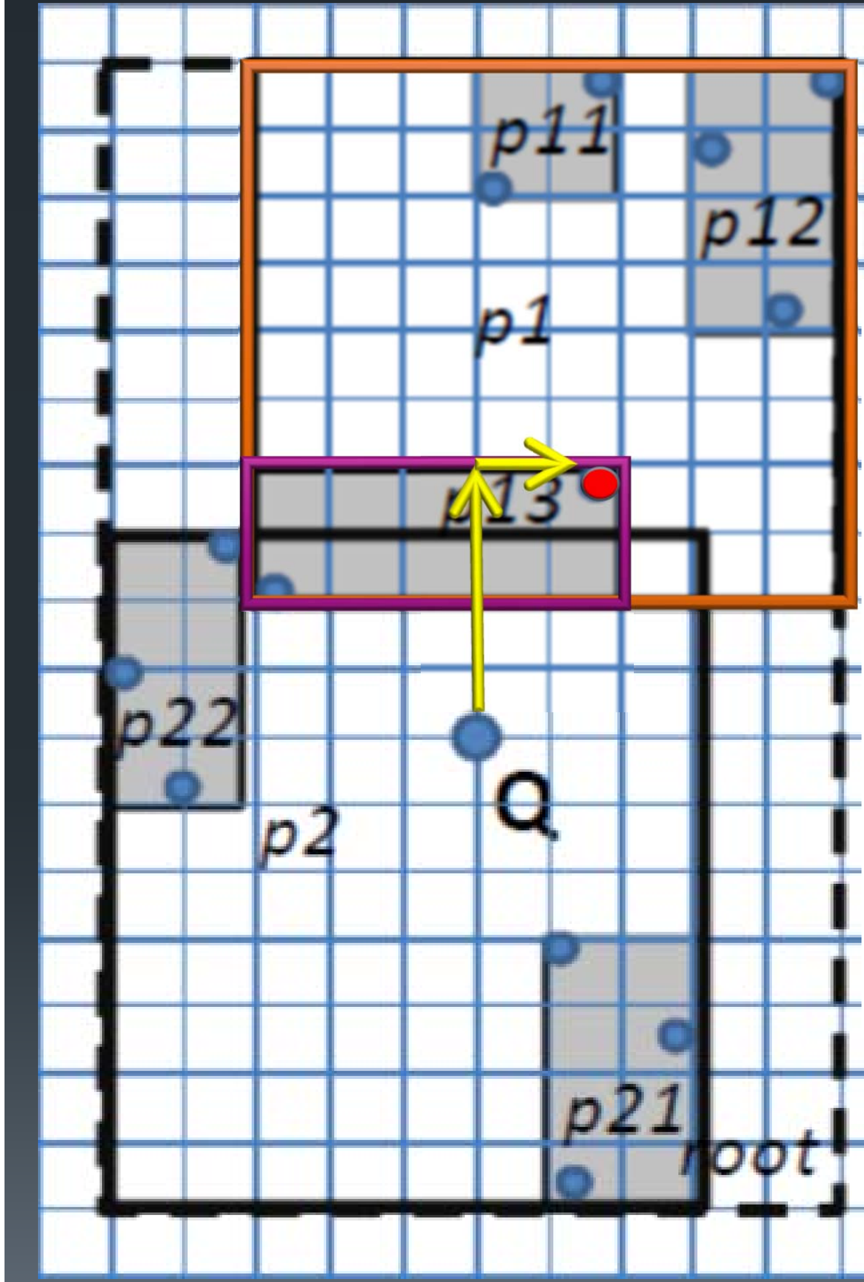
stopdist = dist(Q,o1) = 5



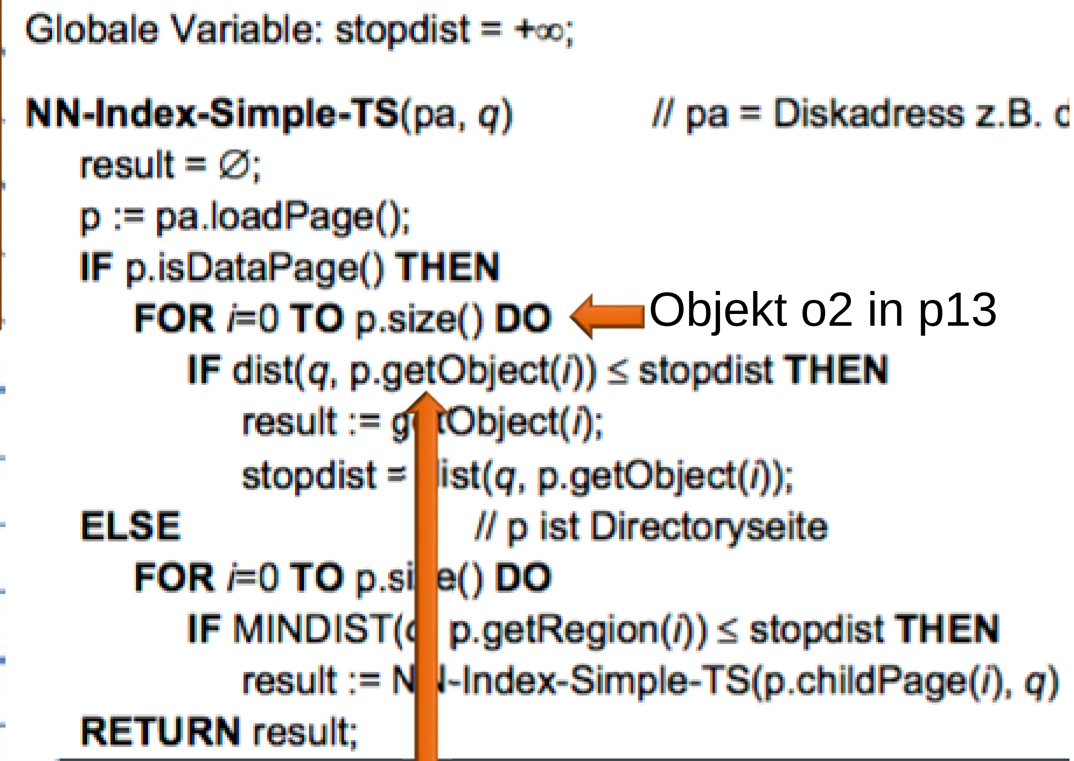
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)           // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                            // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Objekt o1 in p13

Besuchte Seiten: p1,p11,p13



stopdist = dist(Q,o1) = 5

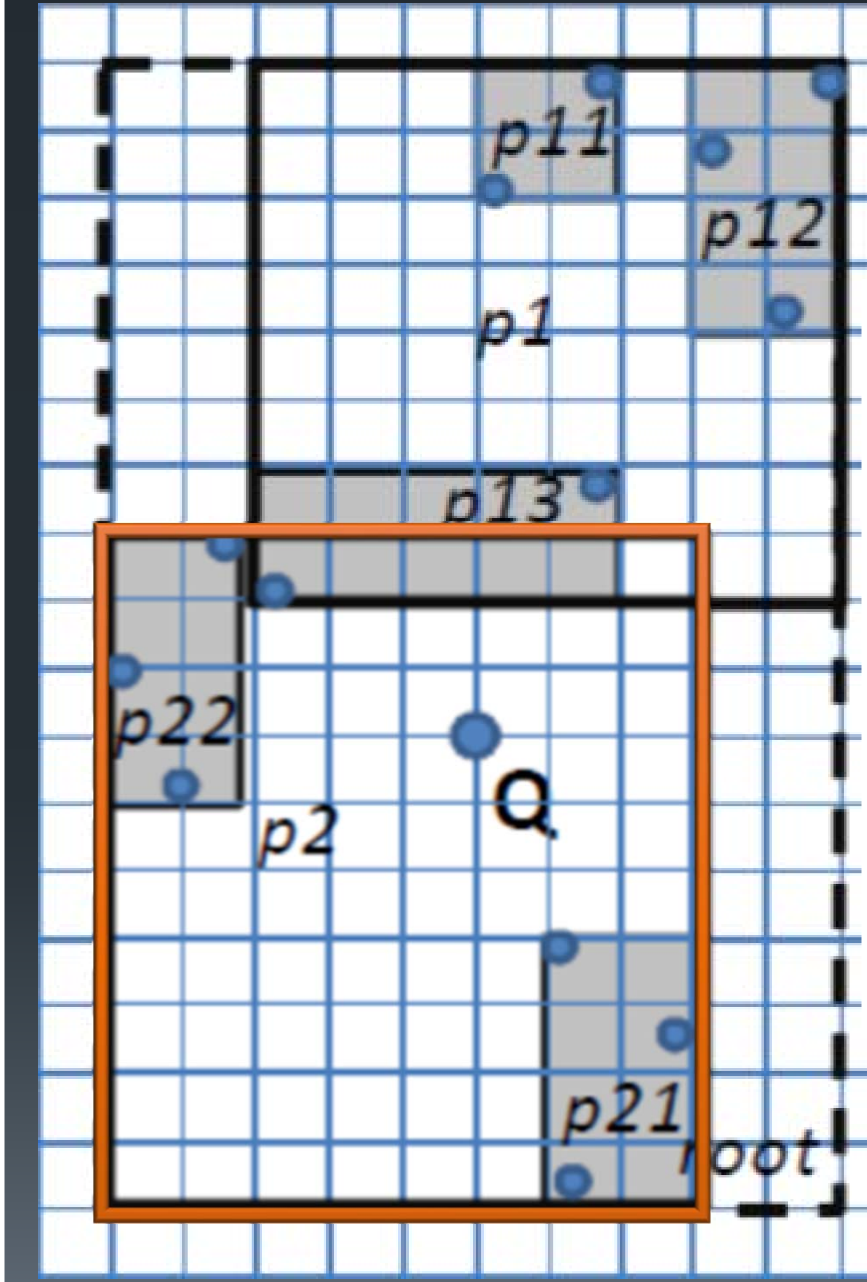


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                              // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

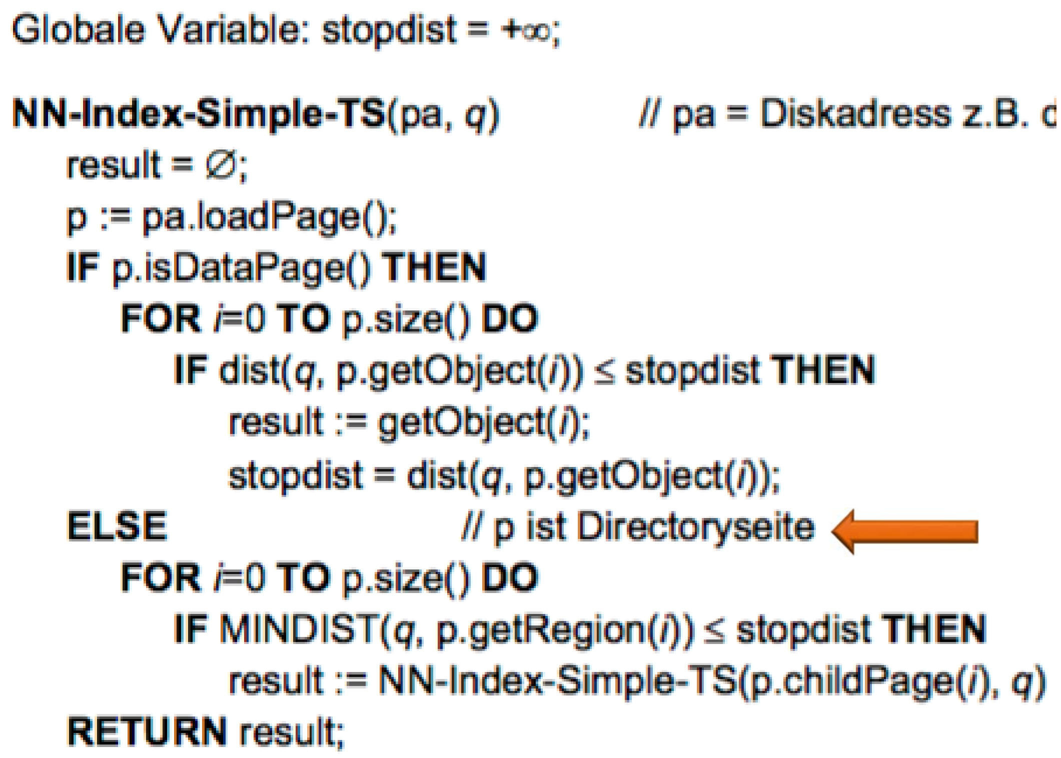
Objekt o2 in p13

dist(Q,o2) = 6 <= 5?
Nein → true drop

Besuchte Seiten: p1,p11,p13,p2

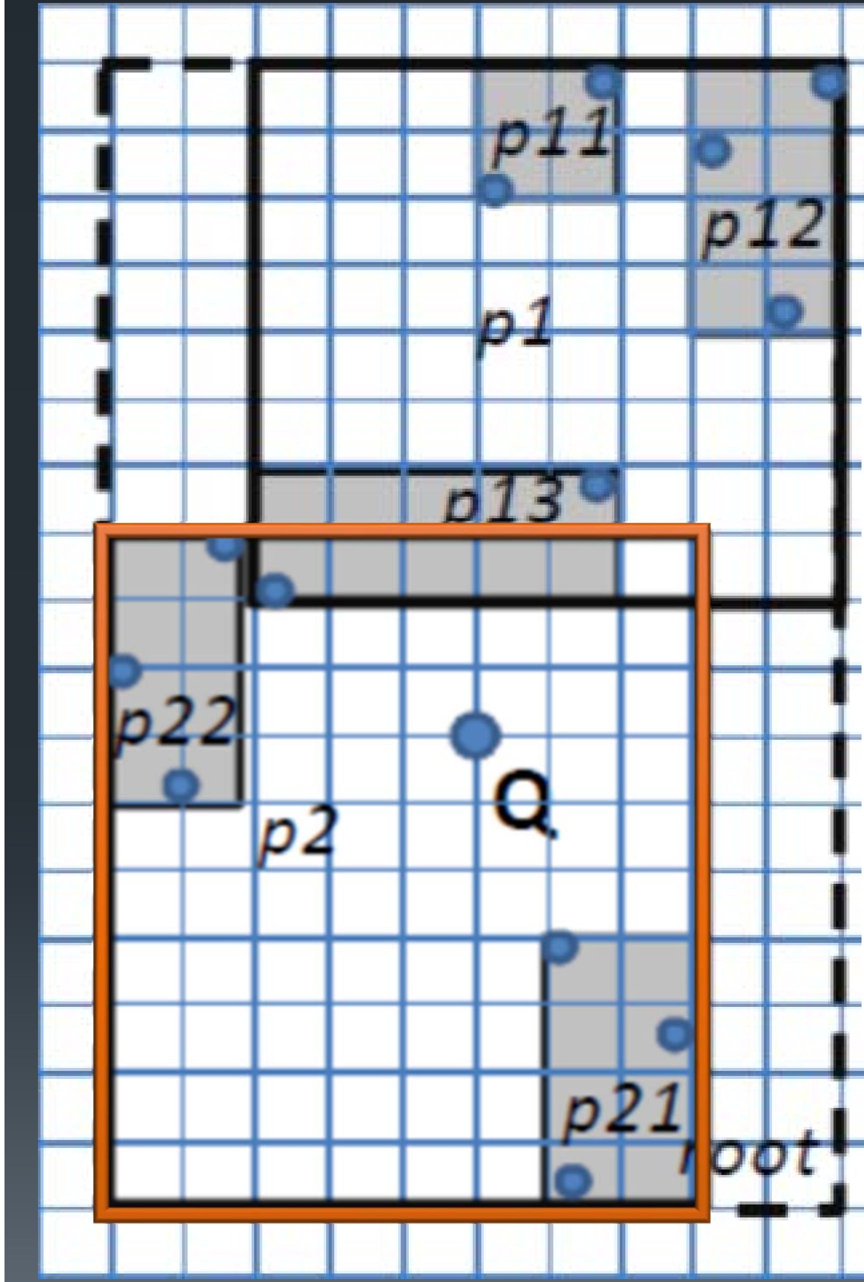


stopdist = dist(Q,o1) = 5

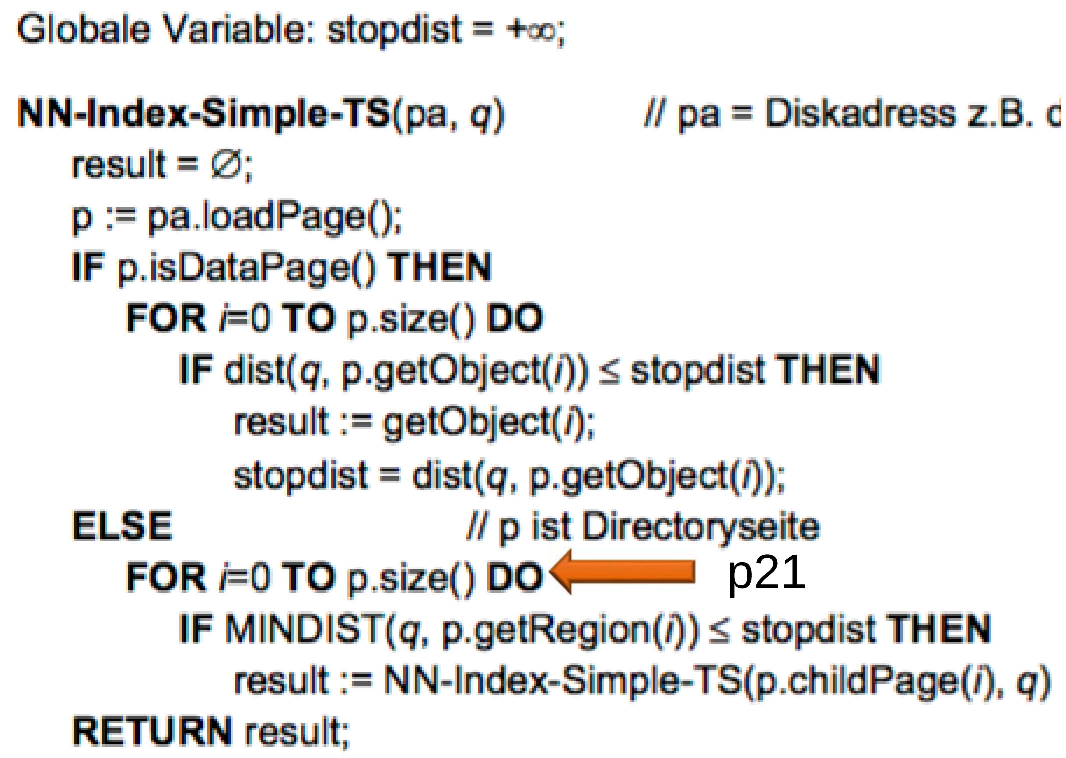


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Besuchte Seiten: p1,p11,p13,p2



stopdist = dist(Q,o1) = 5

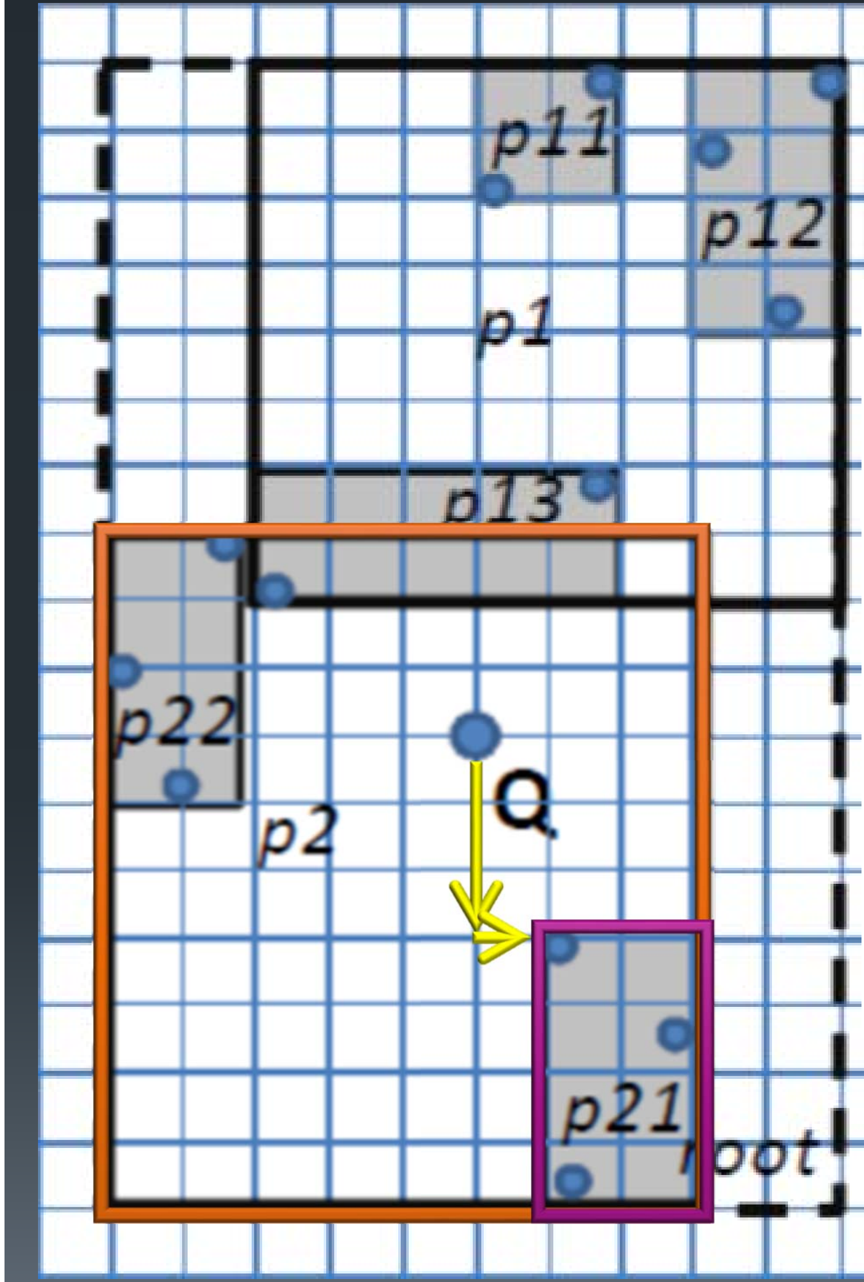


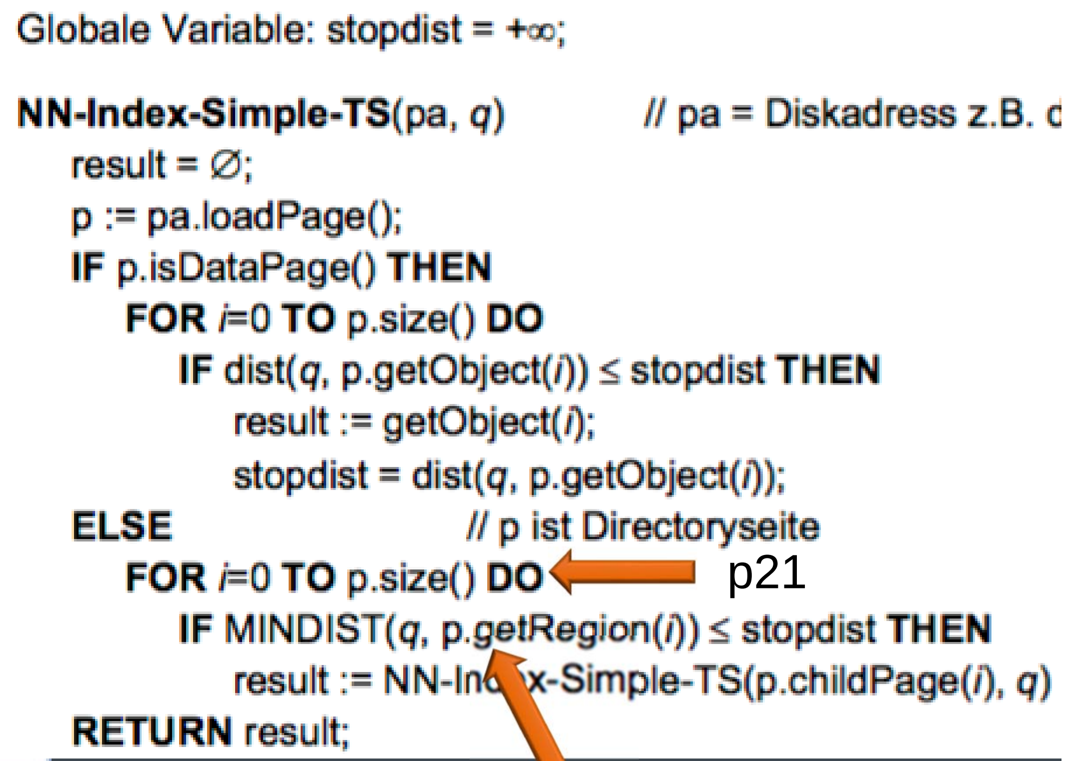
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

p21

Besuchte Seiten: p1,p11,p13,p2

stopdist = dist(Q,o1) = 5



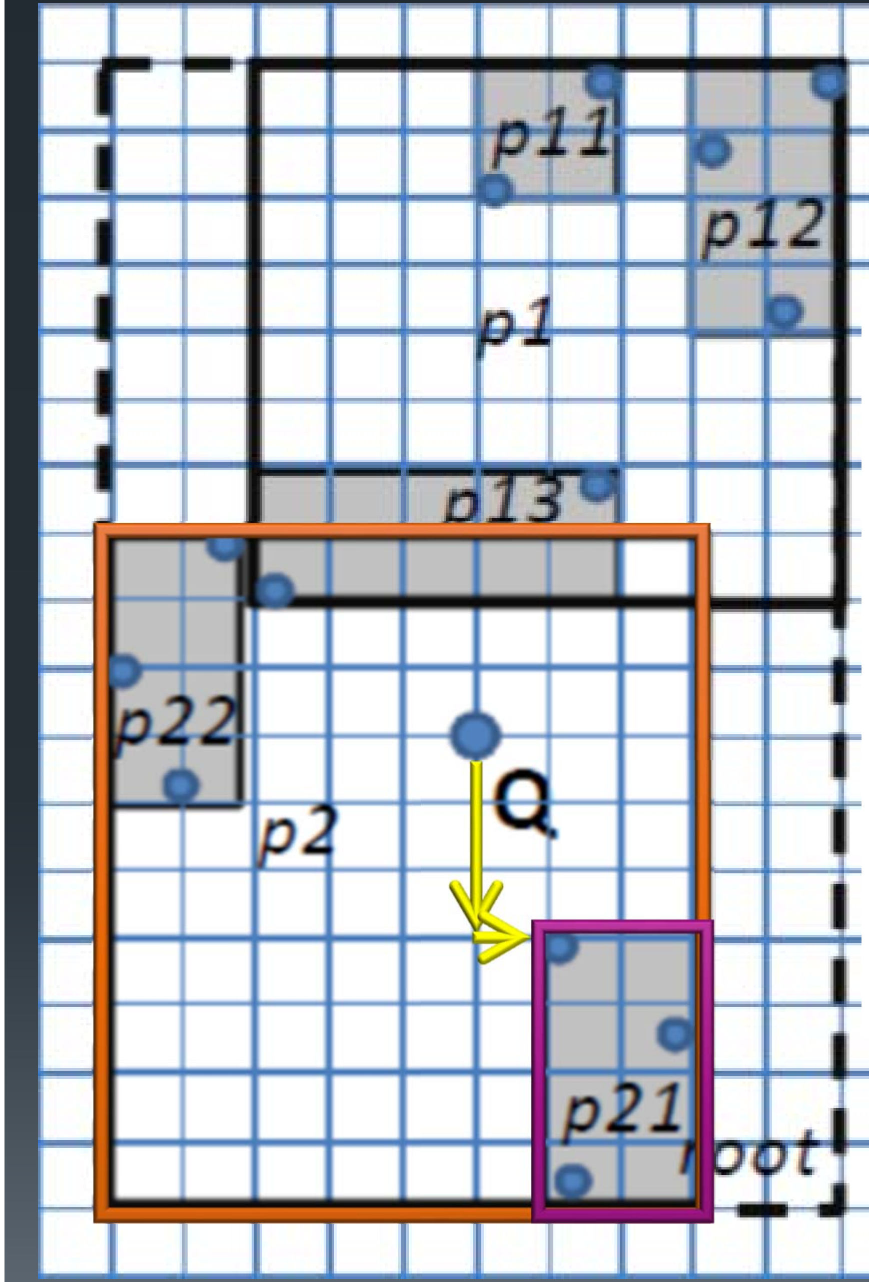


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

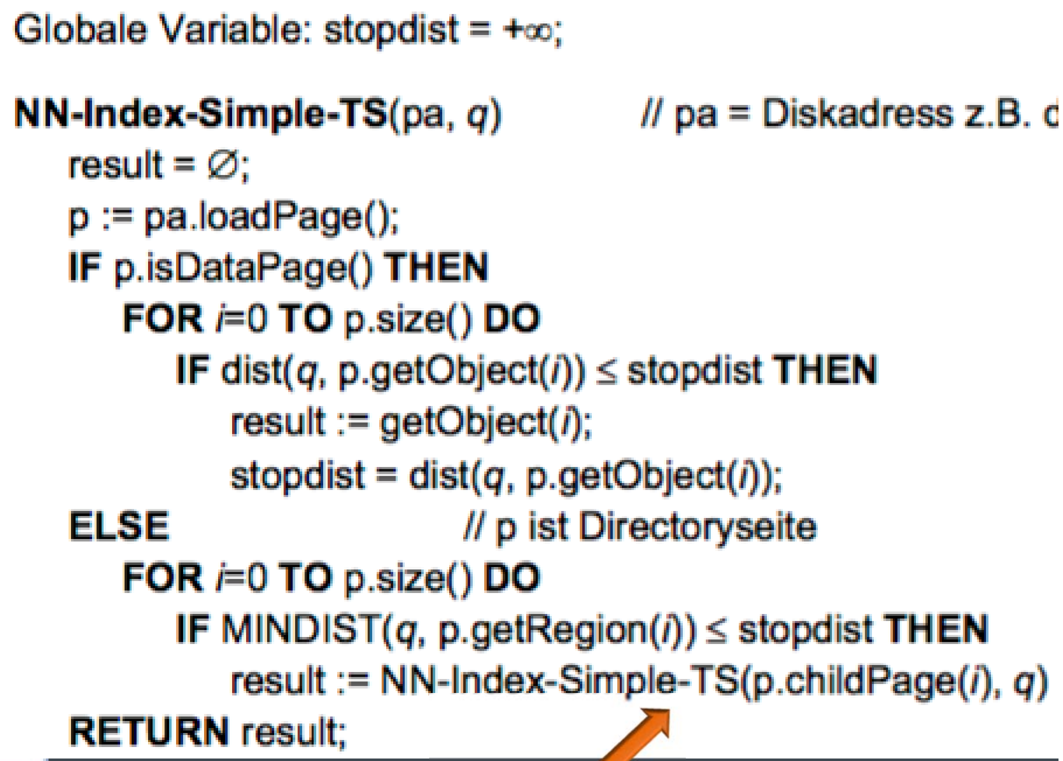
p21

MINDIST(Q,p21)=4 <= 5? → ja!

Besuchte Seiten: p1,p11,p13,p2,p21



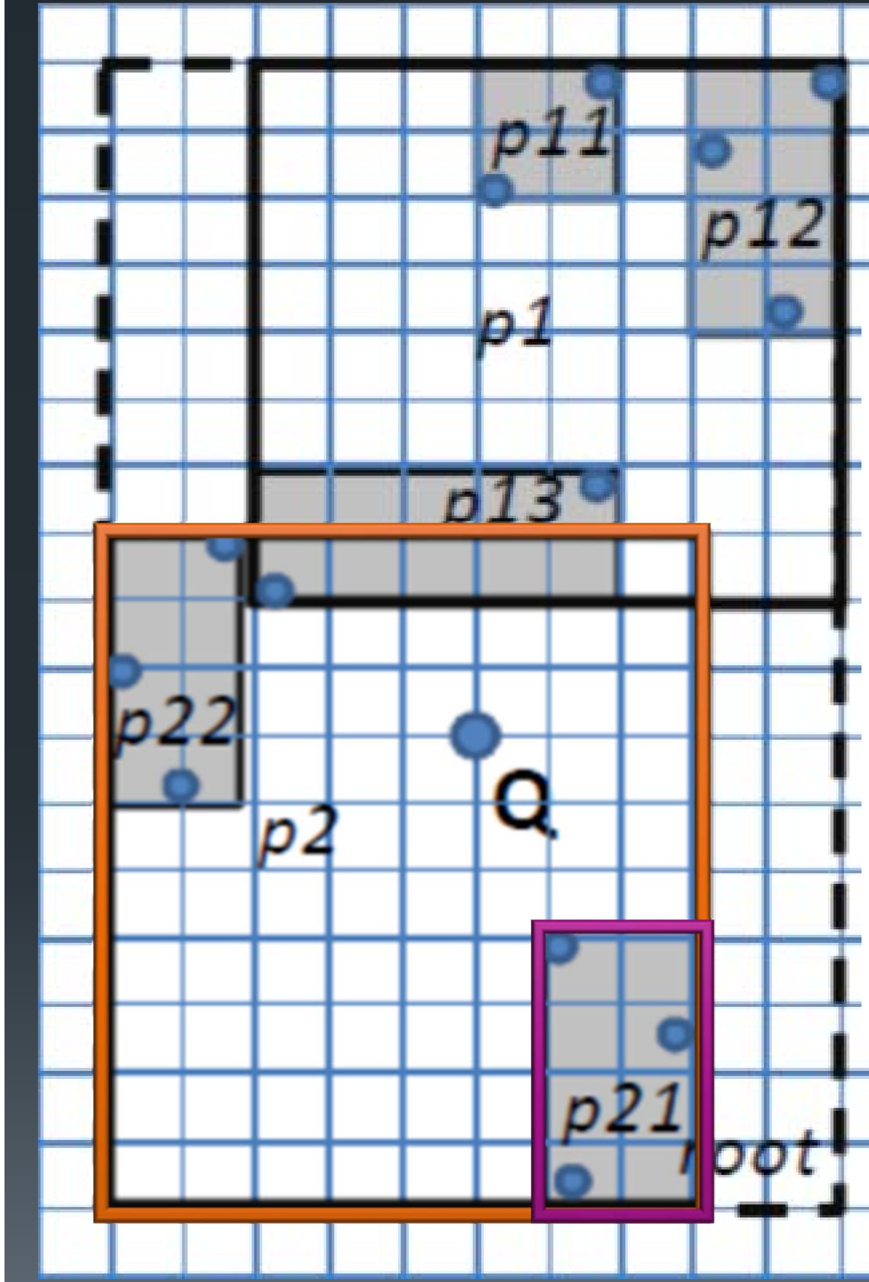
stopdist = dist(Q,o1) = 5



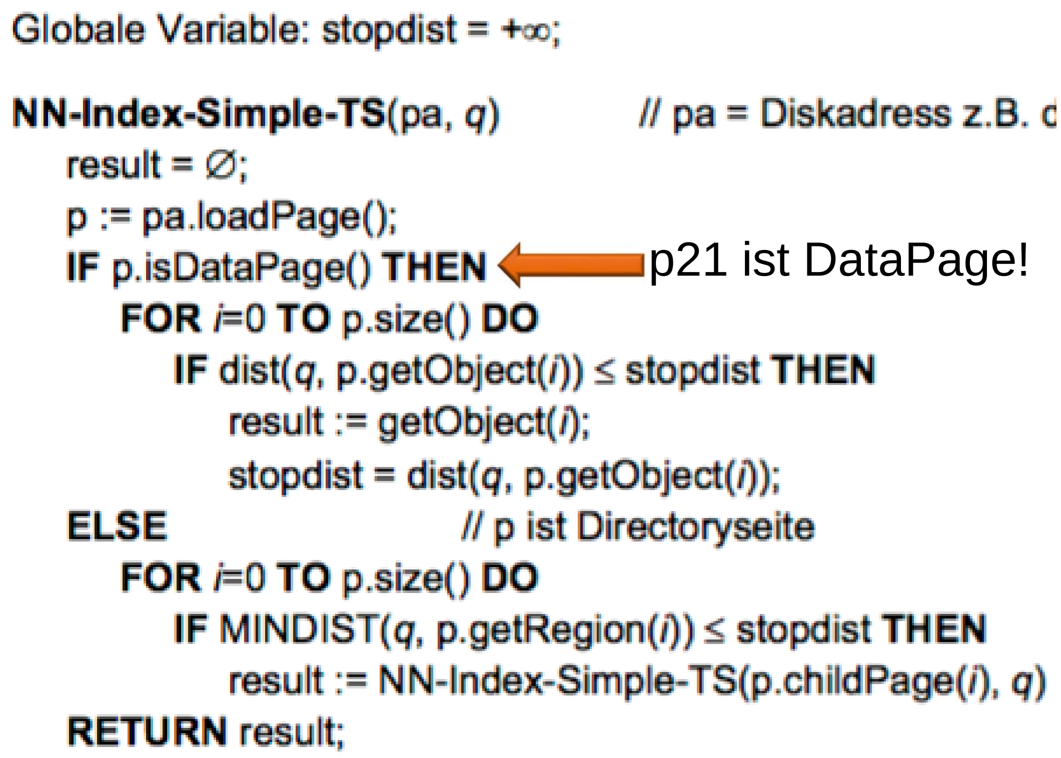
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)             // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

rekursiver Aufruf

Besuchte Seiten: p1,p11,p13,p2,p21



stopdist = dist(Q,o1) = 5



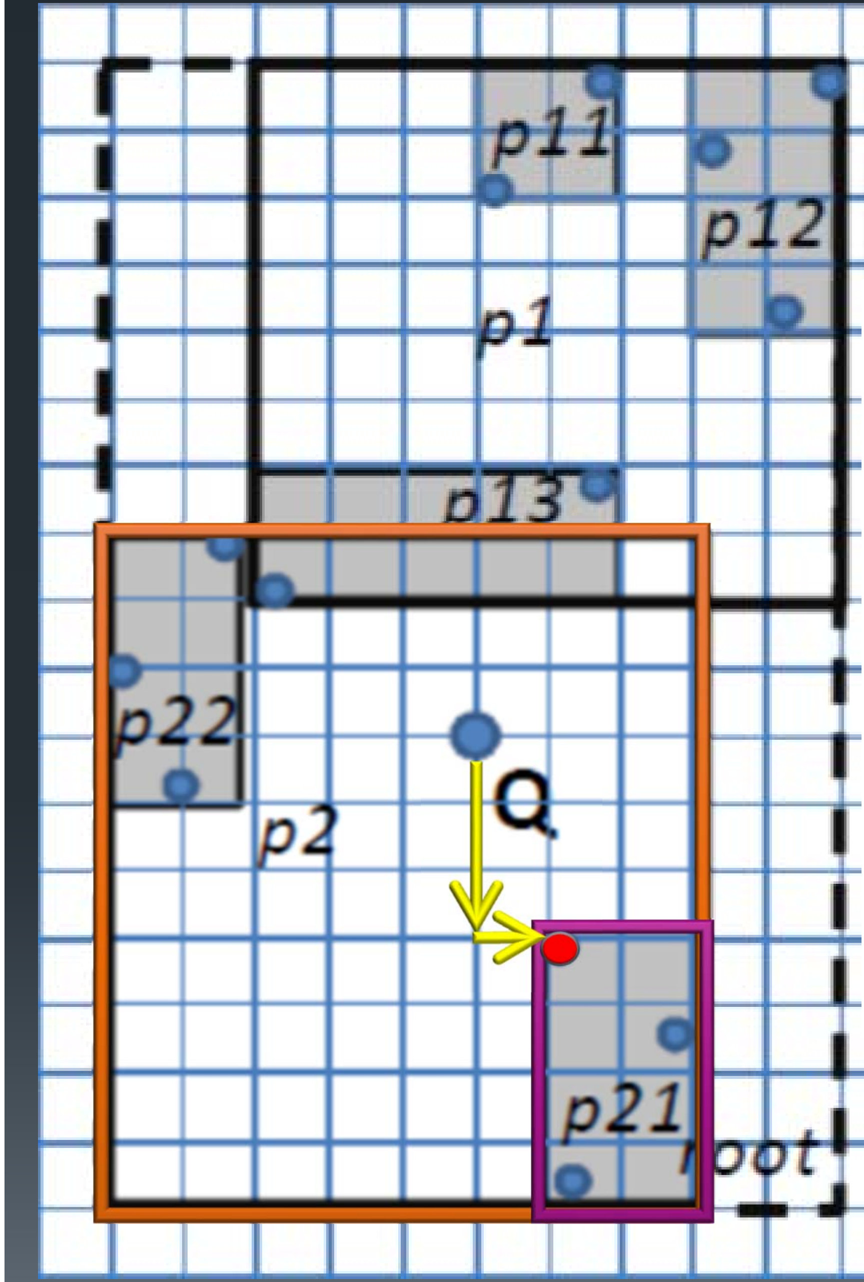
Globale Variable: stopdist = +∞;

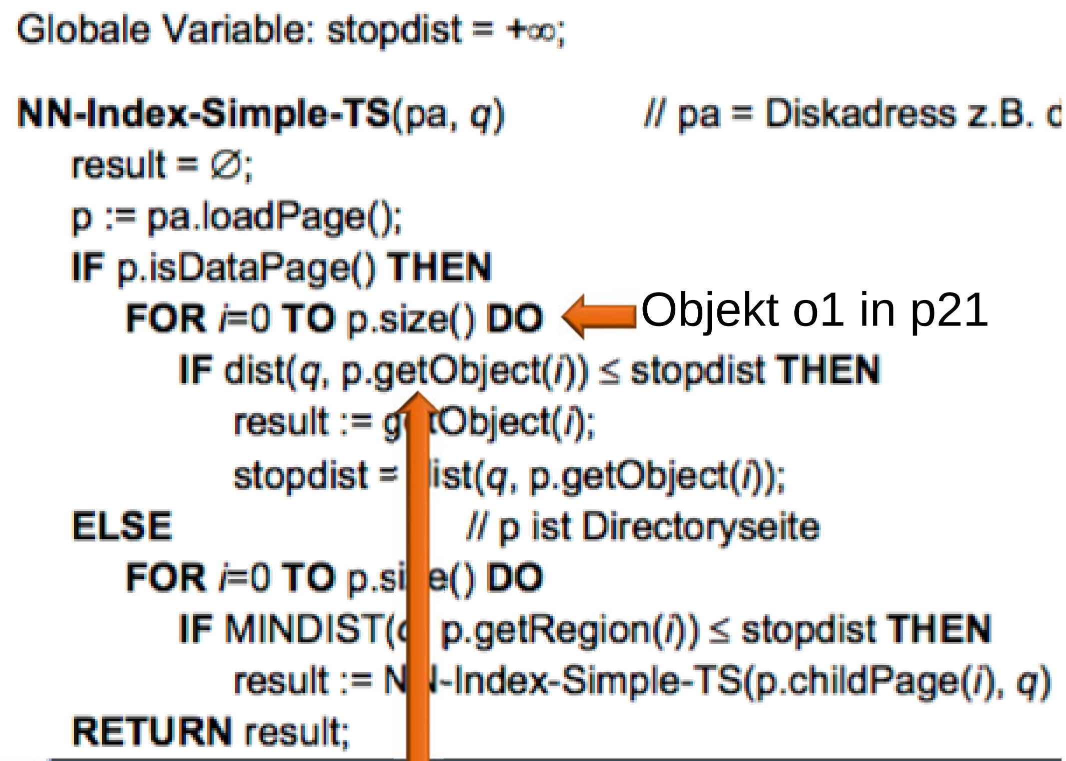
```
NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                         // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

p21 ist DataPage!

Besuchte Seiten: p1,p11,p13,p2,p21

stopdist = dist(Q,o1) = 5



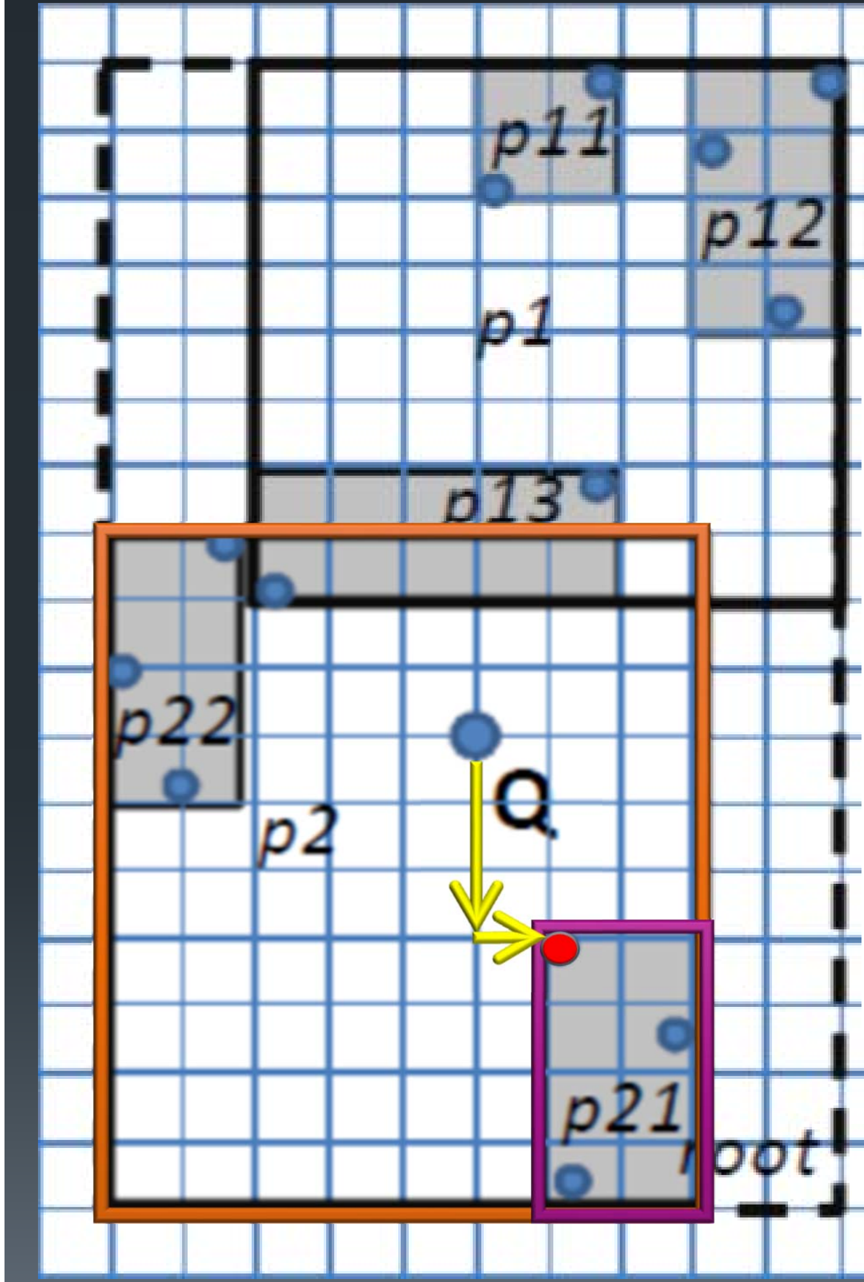


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

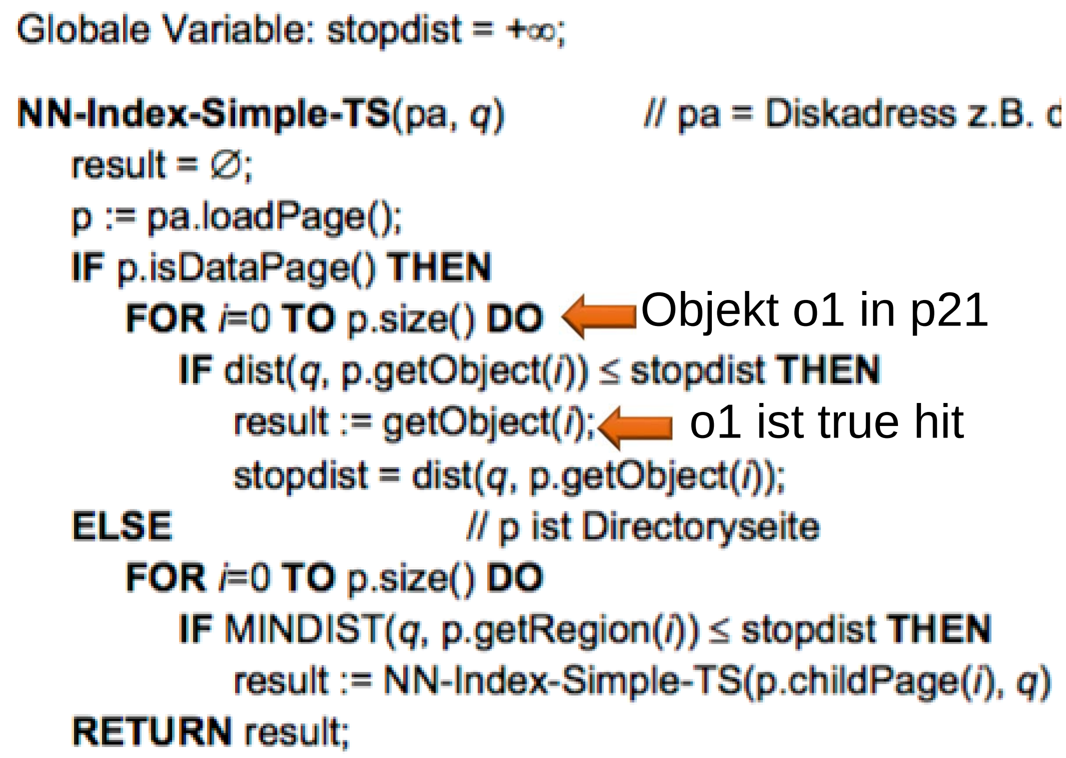
Objekt o1 in p21

dist(Q,o1) = 4 <= 5

Besuchte Seiten: p1,p11,p13,p2,p21



stopdist = dist(Q,o1) = 5

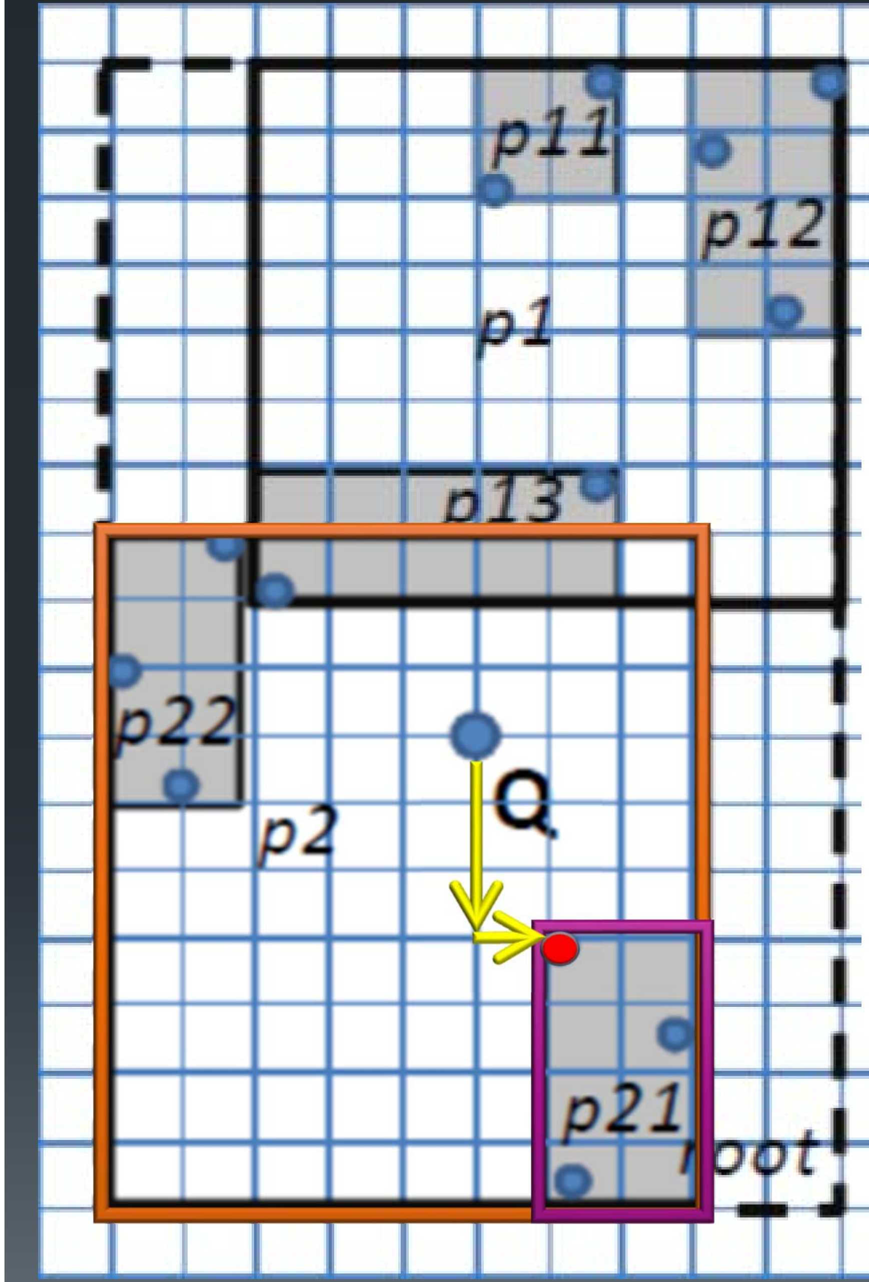


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
   result = ∅;
   p := pa.loadPage();
   IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
         IF dist(q, p.getObject(i)) ≤ stopdist THEN
            result := getObject(i);
            stopdist = dist(q, p.getObject(i));
   ELSE                    // p ist Directoryseite
      FOR i=0 TO p.size() DO
         IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
            result := NN-Index-Simple-TS(p.childPage(i), q)
   RETURN result;
```

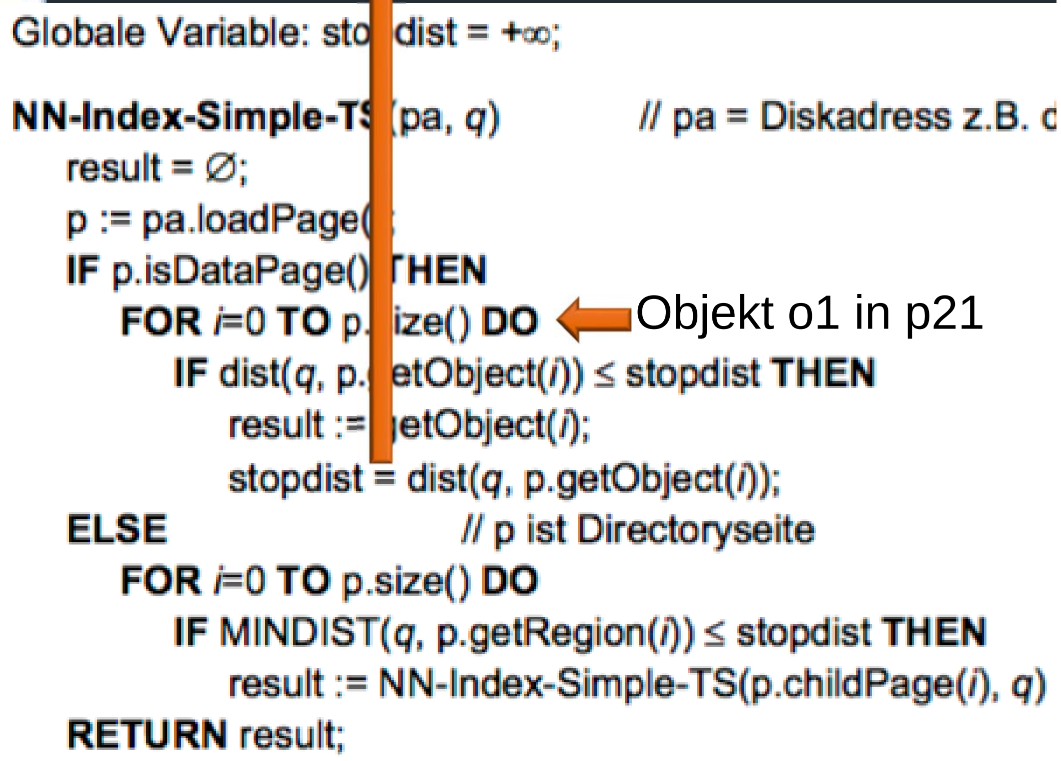
Objekt o1 in p21

o1 ist true hit

Besuchte Seiten: p1,p11,p13,p2,p21



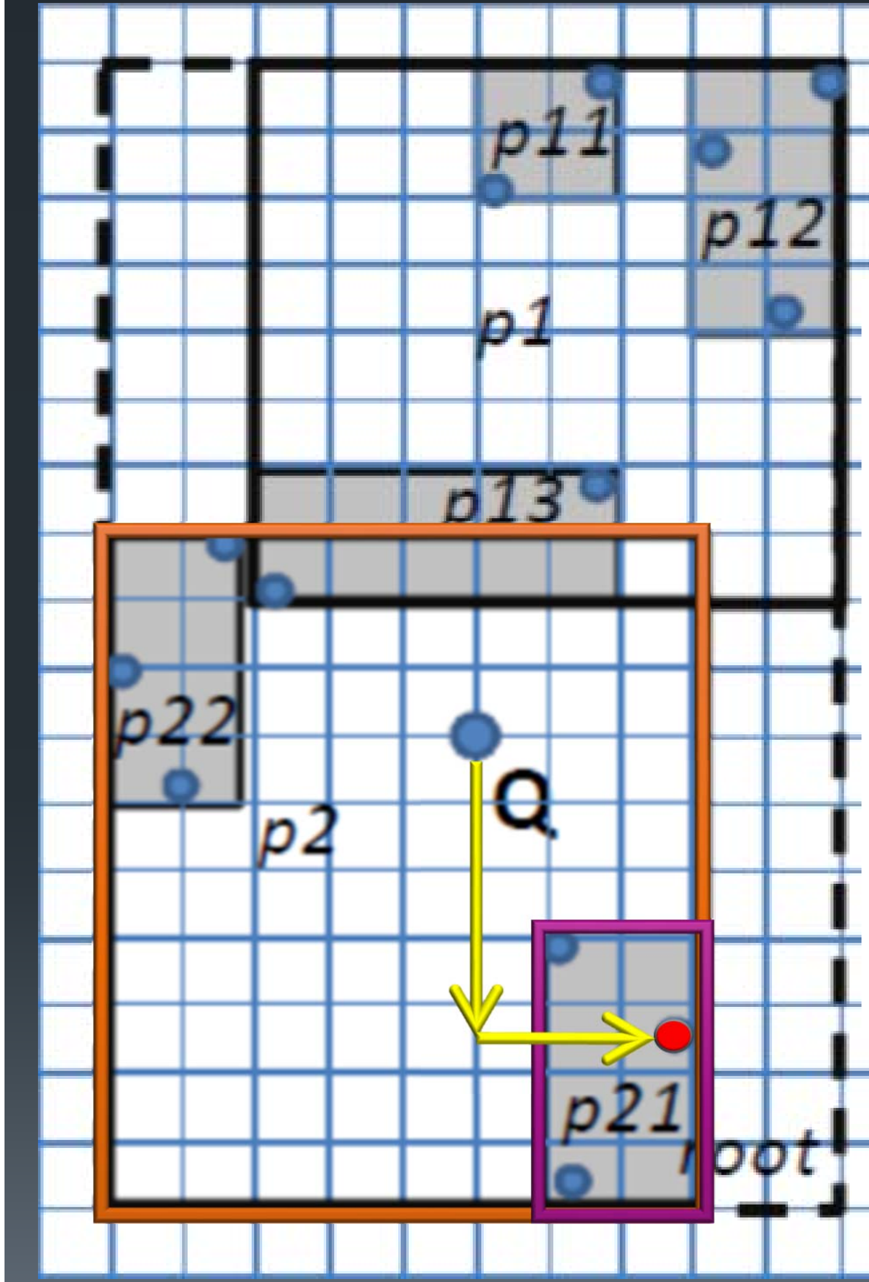
stopdist = dist(Q,o1) = 4



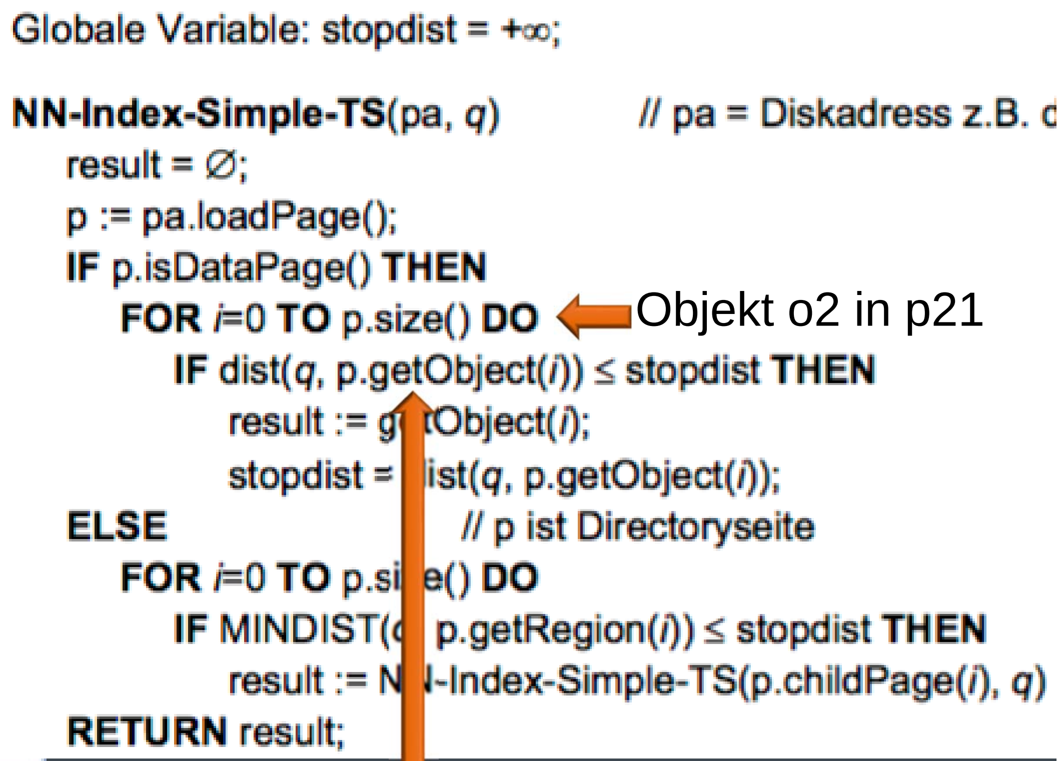
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                              // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Objekt o1 in p21

Besuchte Seiten: p1,p11,p13,p2,p21



stopdist = dist(Q,o1) = 4

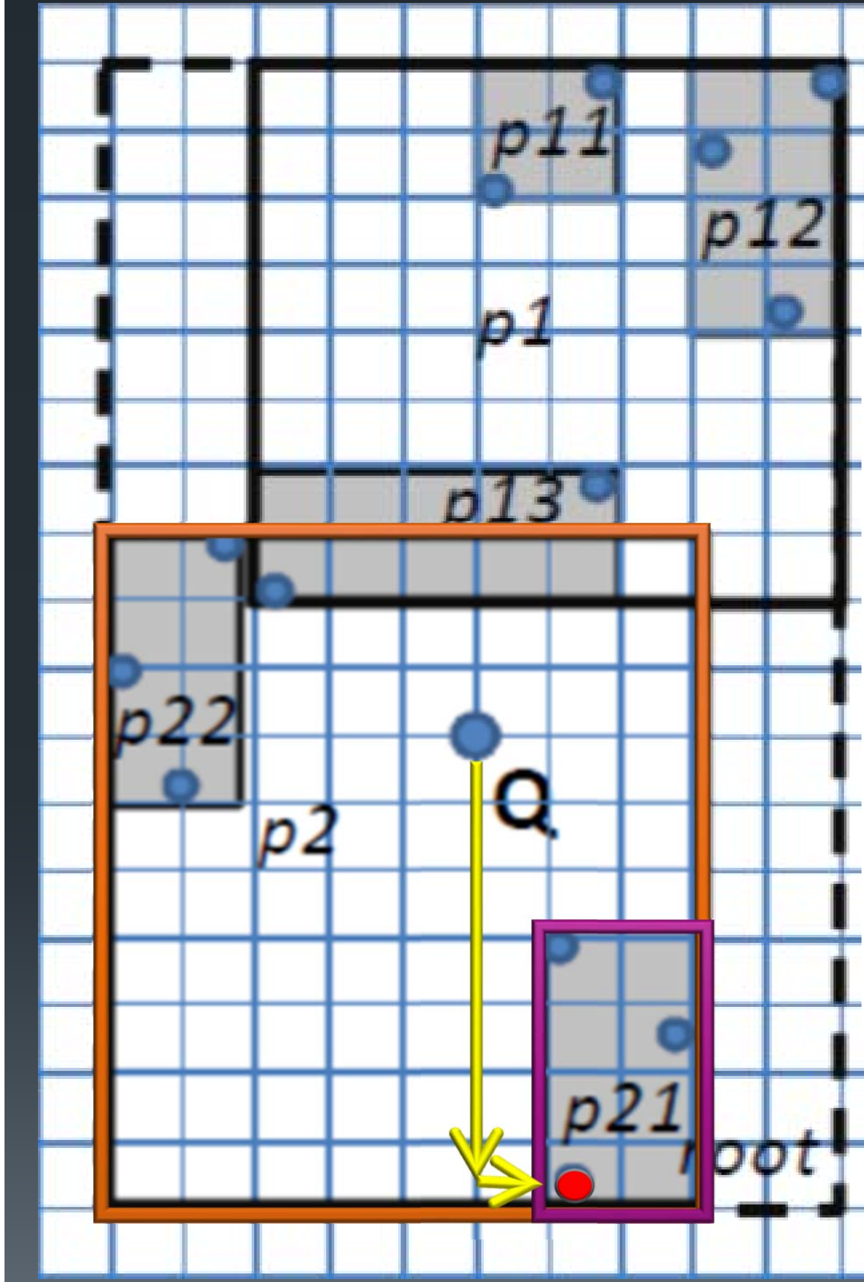


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

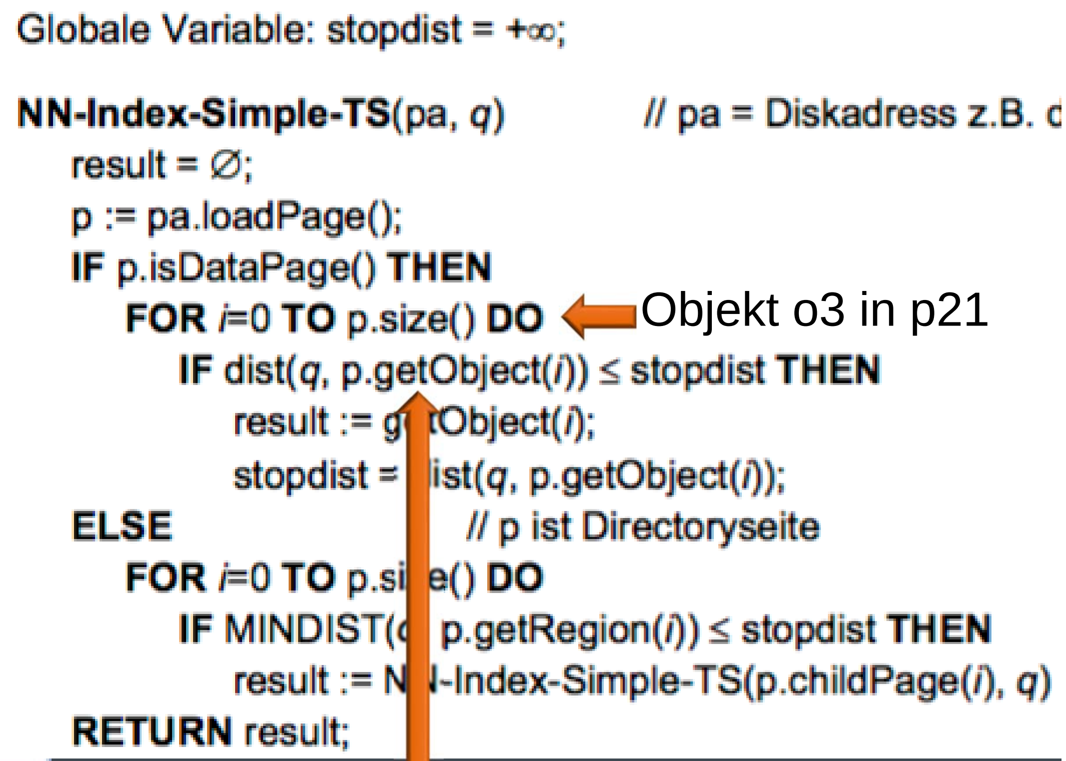
Objekt o2 in p21

dist(Q,o2) = 7.5 <= 4?
Nein → true drop

Besuchte Seiten: p1,p11,p13,p2,p21



stopdist = dist(Q,o1) = 4

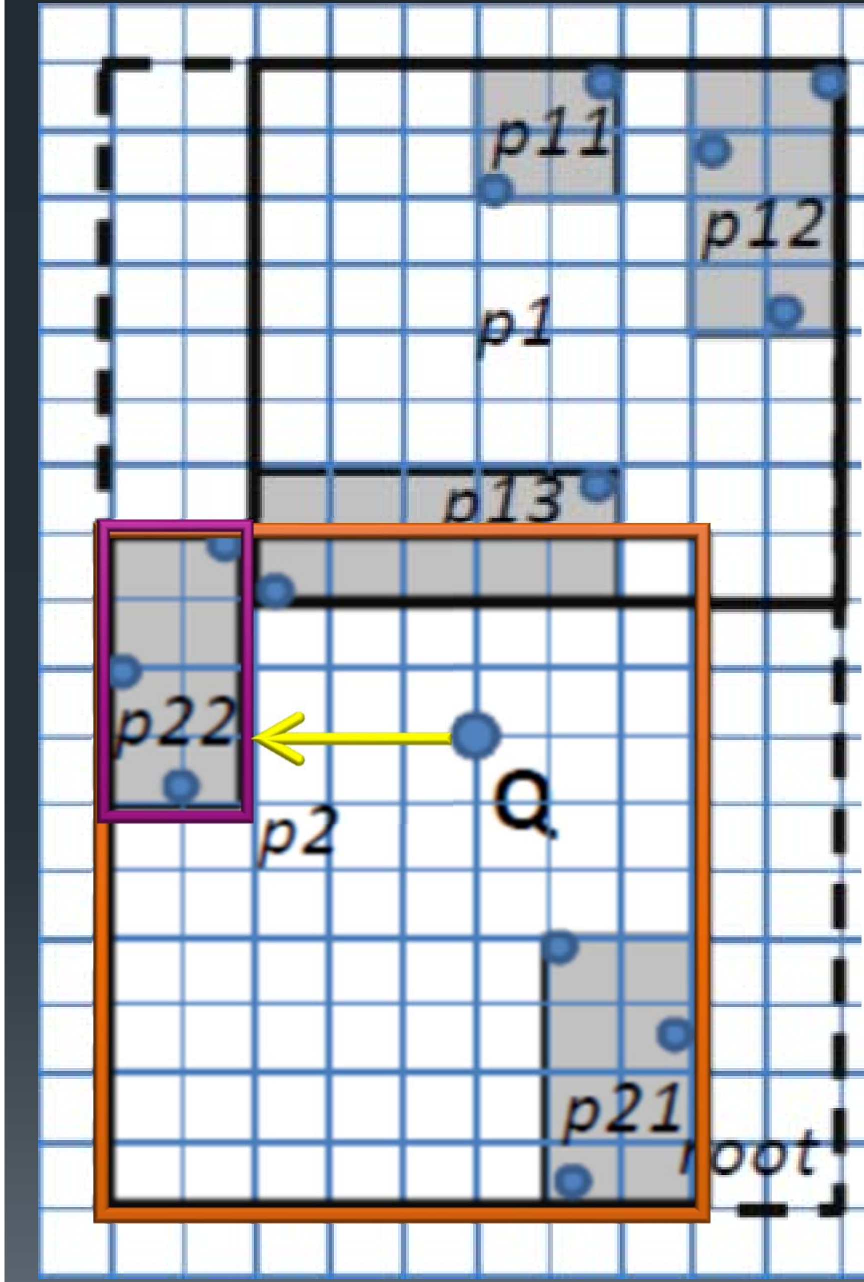


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                              // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

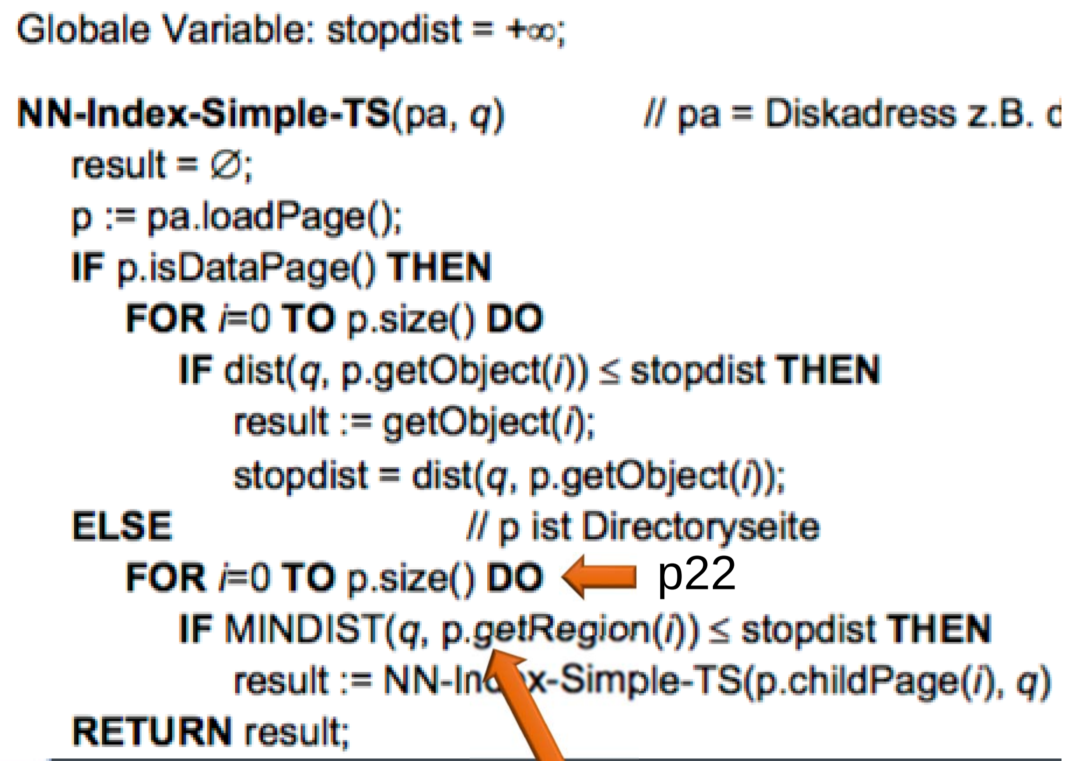
Objekt o3 in p21

dist(Q,o3) = 8.5 <= 4?
Nein → true drop

Besuchte Seiten: p1,p11,p13,p2,p21



stopdist = dist(Q,o1) = 4

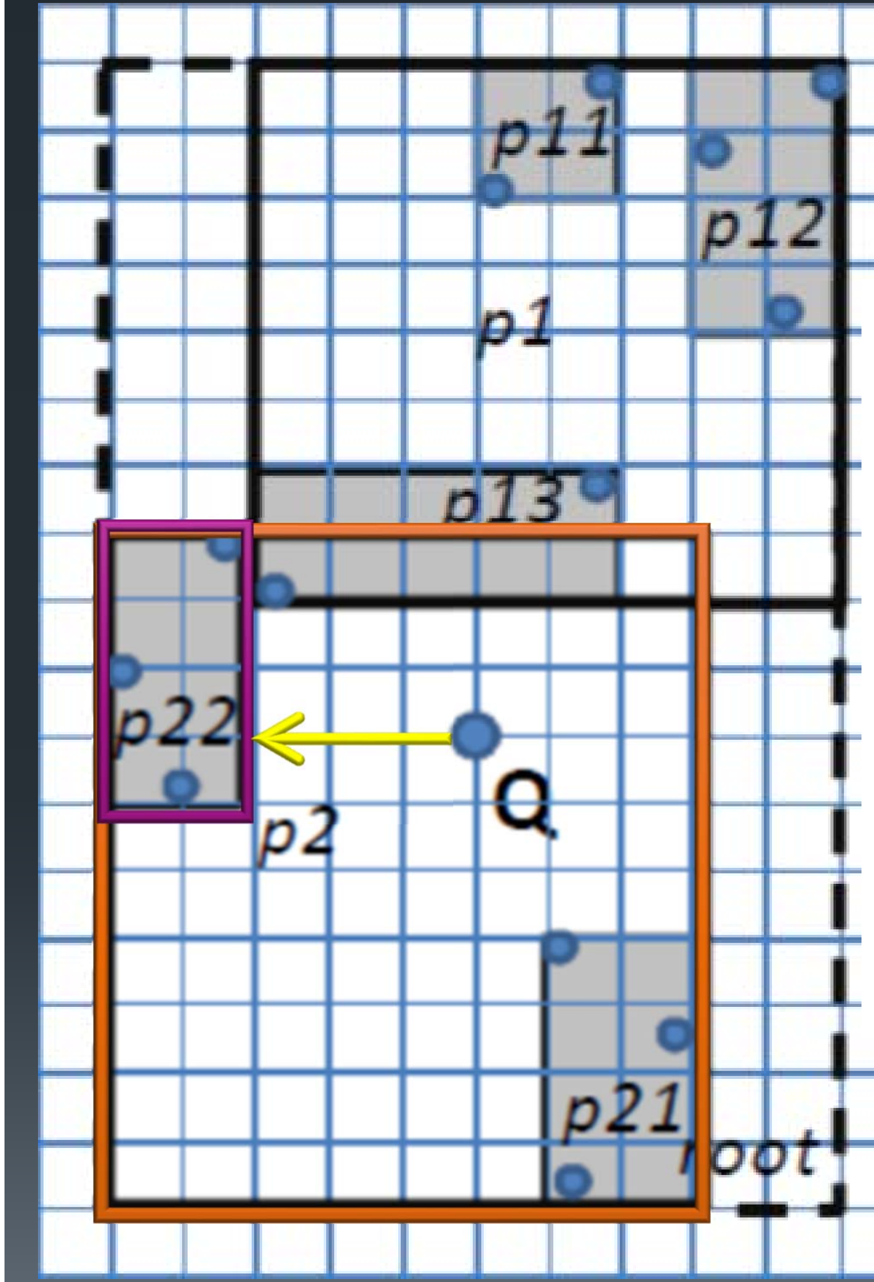


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                           // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

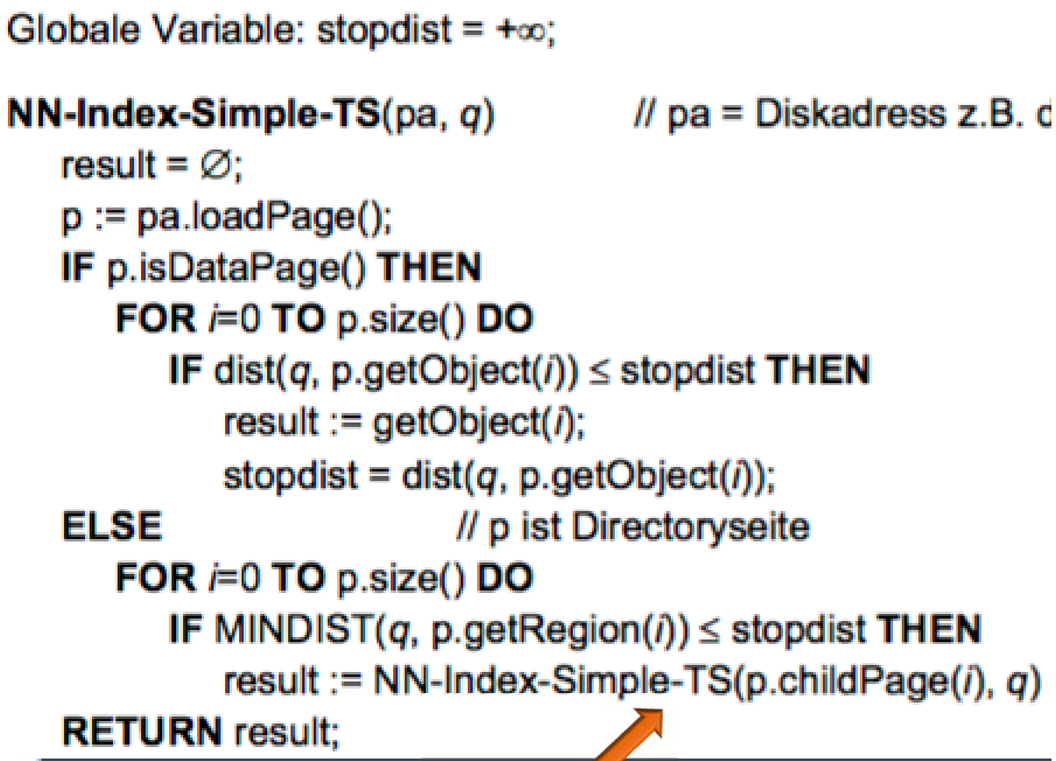
p22

MINDIST(Q,p22)=3 <= 4? → ja!

Besuchte Seiten: p1,p11,p13,p2,p21,p22



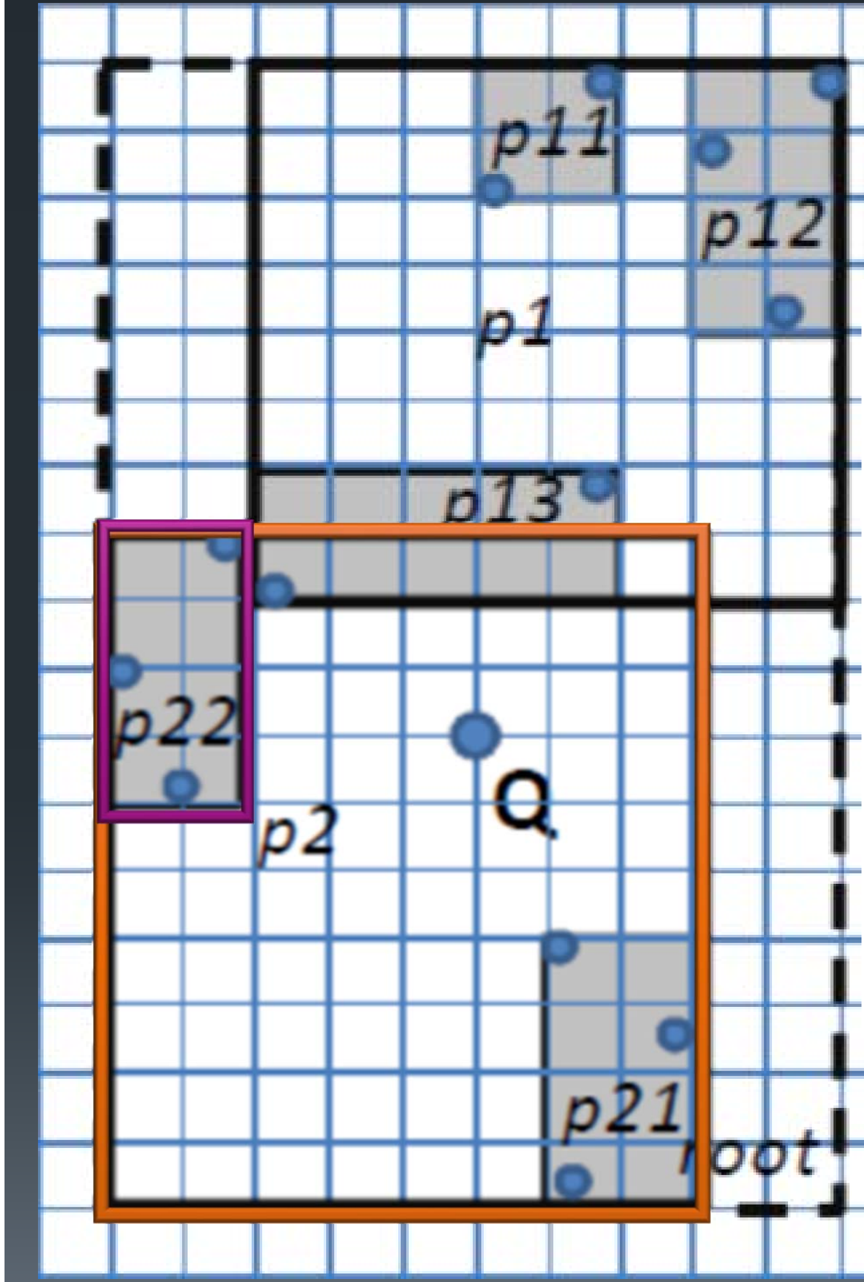
stopdist = dist(Q,o1) = 4



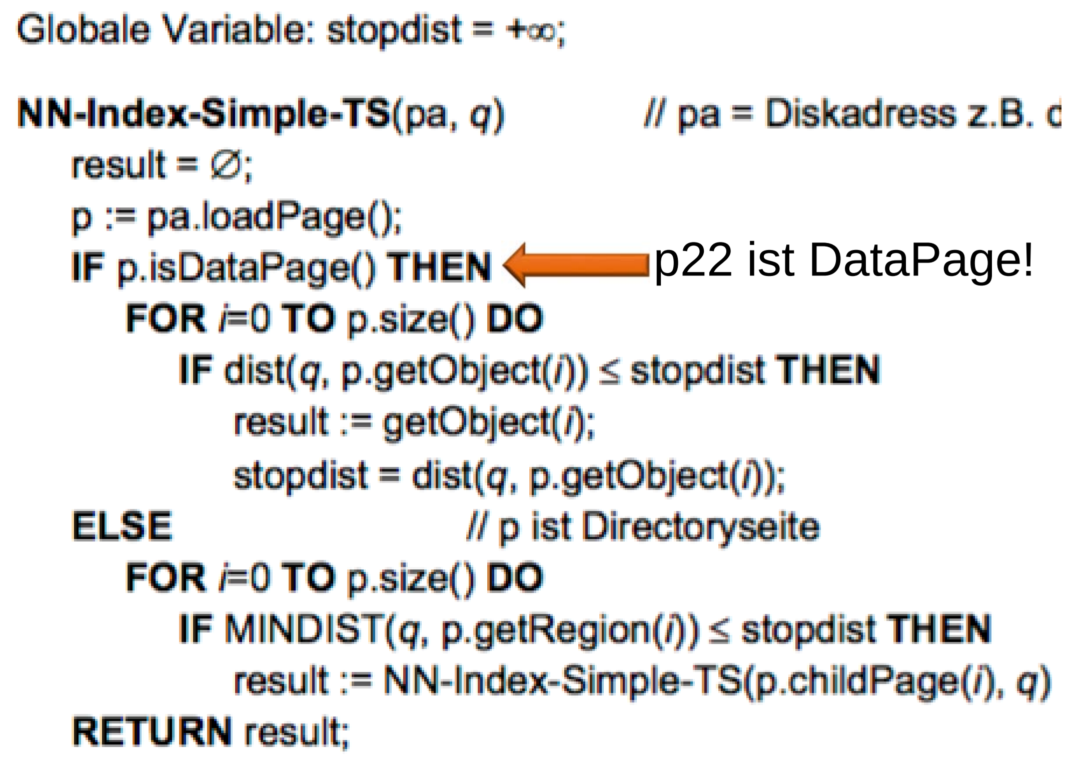
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                          // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

rekursiver Aufruf

Besuchte Seiten: p1,p11,p13,p2,p21,p22



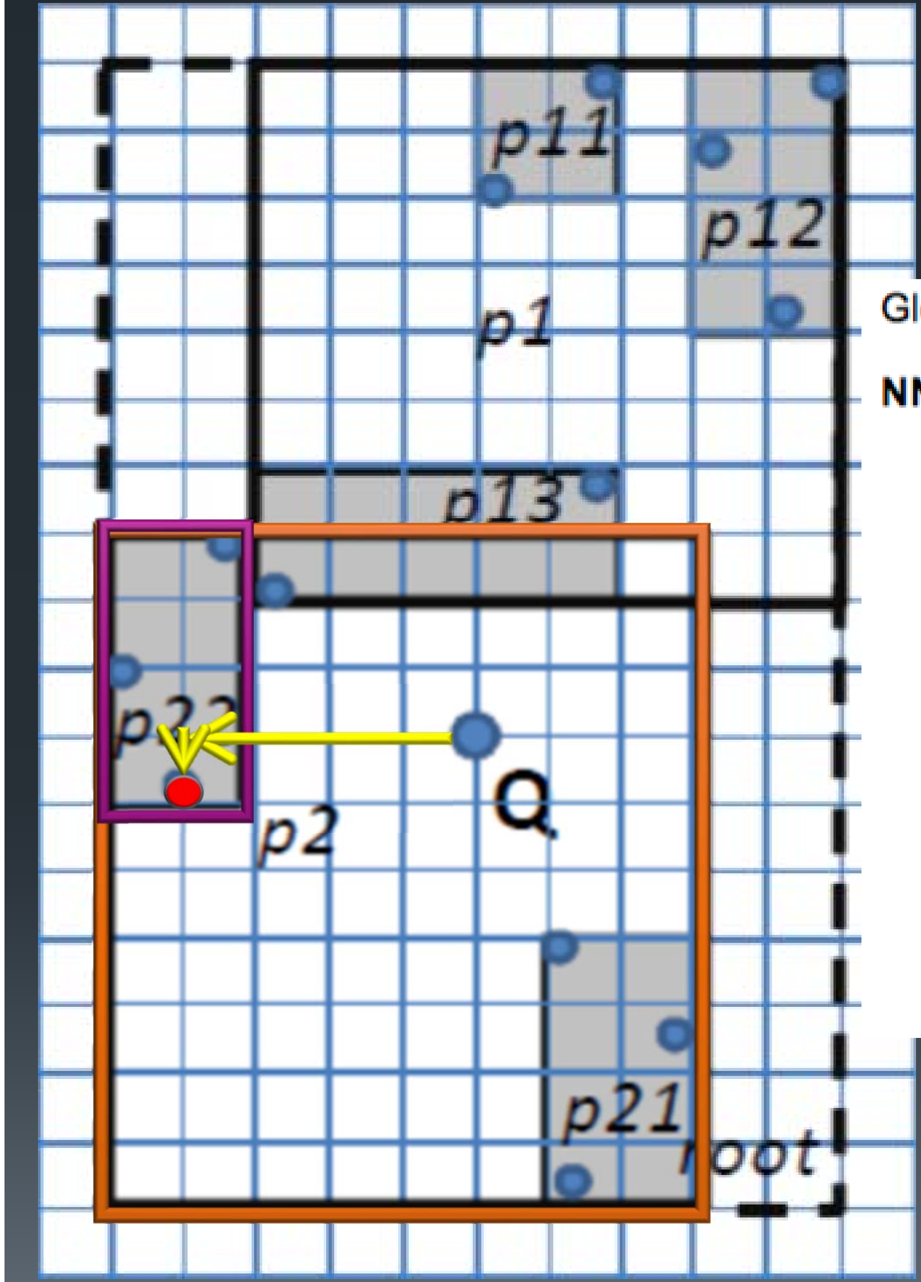
stopdist = dist(Q,o1) = 4



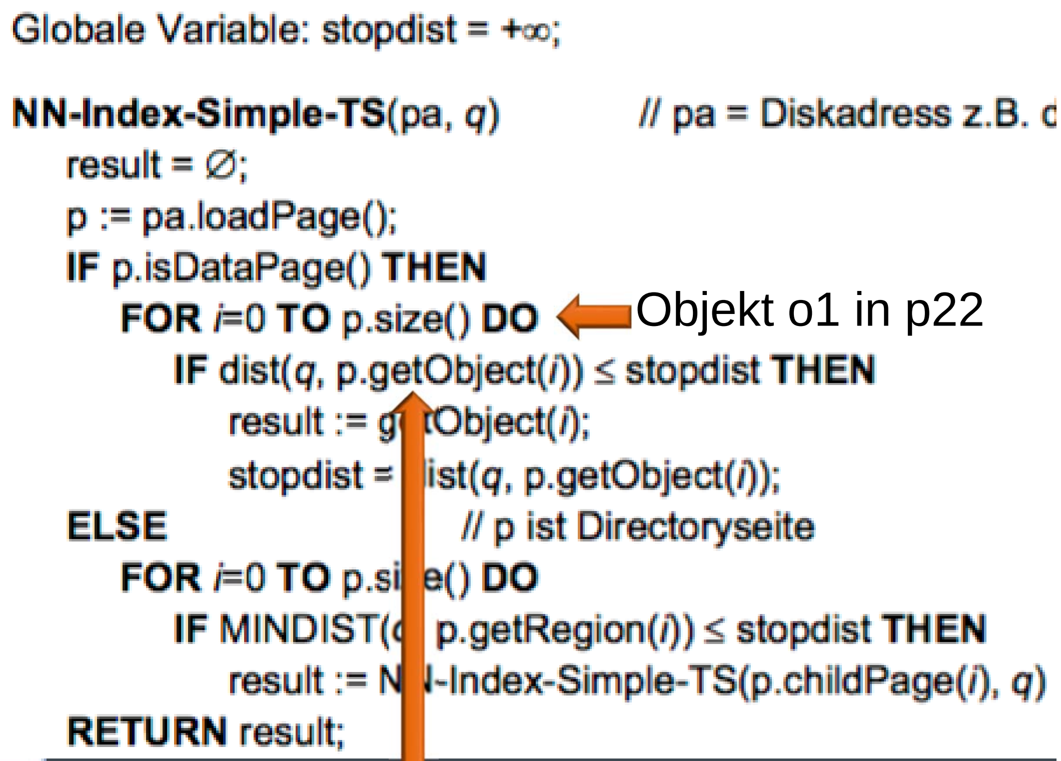
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                        // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

p22 ist DataPage!

Besuchte Seiten: p1,p11,p13,p2,p21,p22



stopdist = dist(Q,o1) = 4

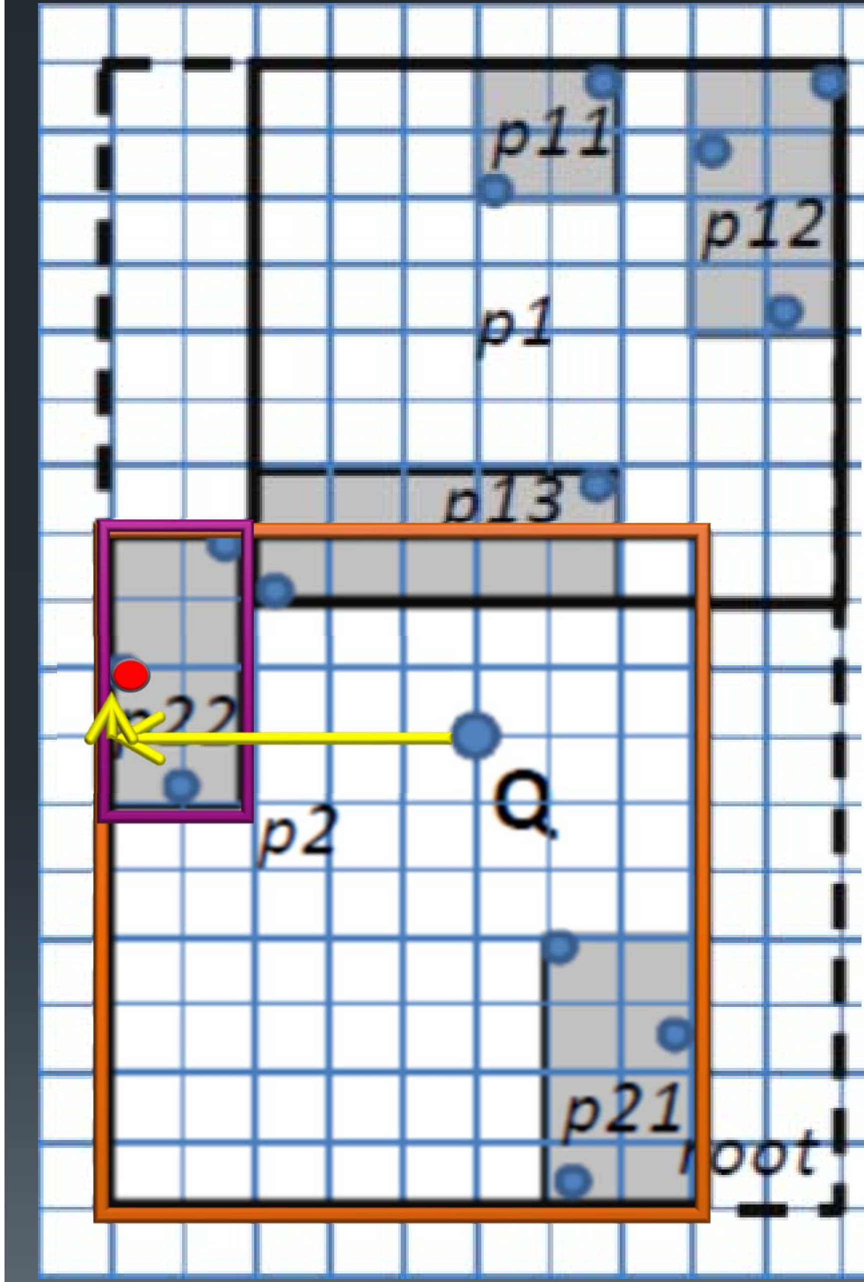


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                                   // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

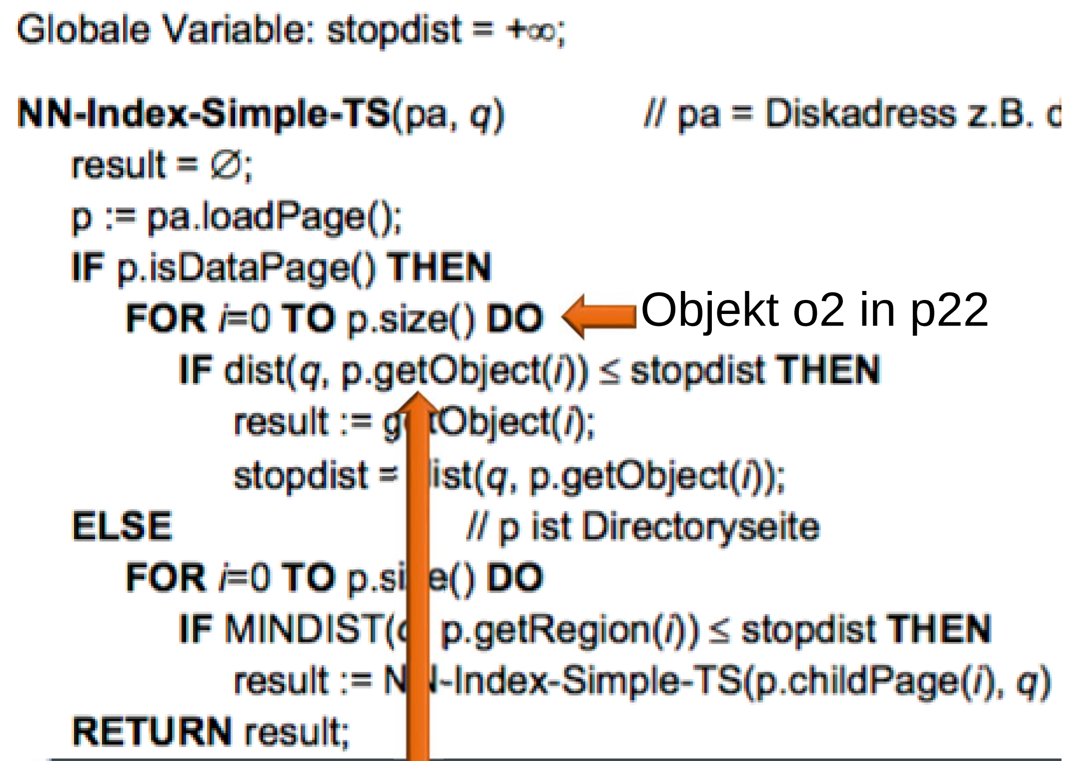
Objekt o1 in p22

dist(Q,o1) = 5 <= 4?
Nein → true drop

Besuchte Seiten: p1,p11,p13,p2,p21,p22



stopdist = dist(Q,o1) = 4

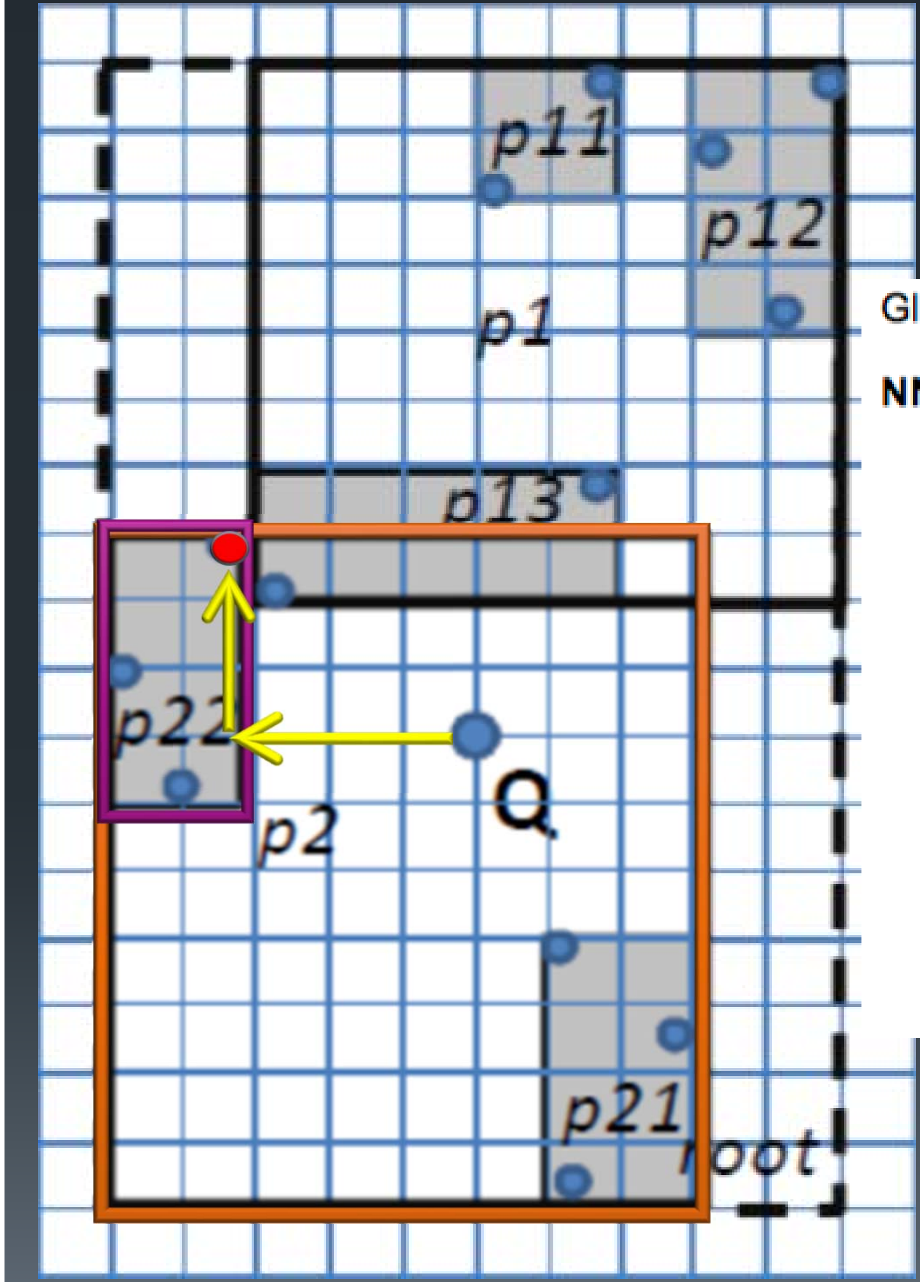


```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)            // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                        // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

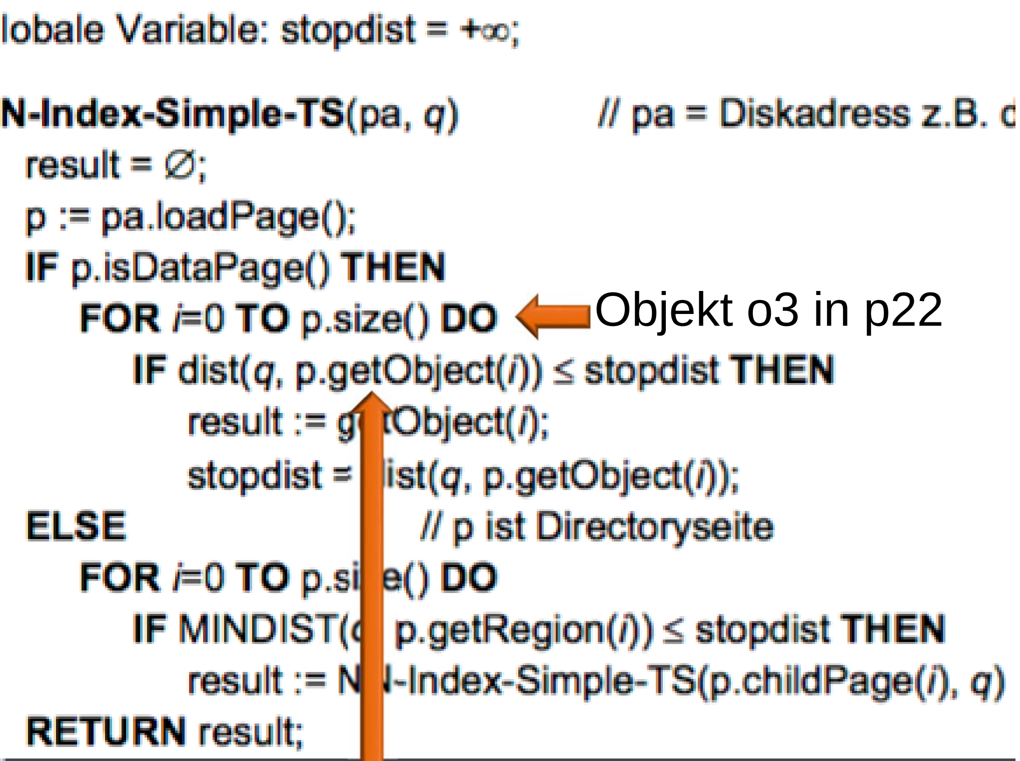
Objekt o2 in p22

dist(Q,o2) = 6 <= 4?
Nein → true drop

Besuchte Seiten: p1,p11,p13,p2,p21,p22



stopdist = dist(Q,o1) = 4



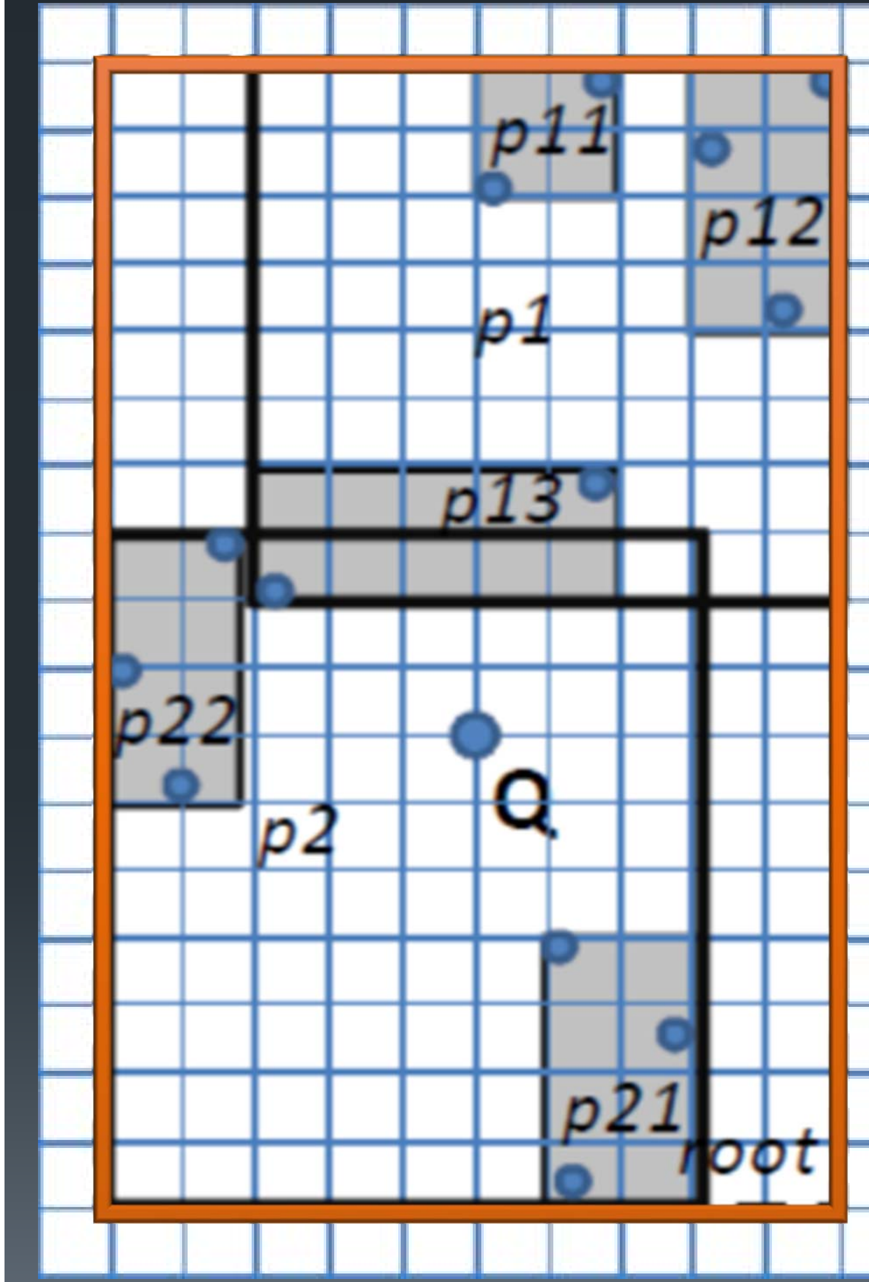
```
Globale Variable: stopdist = +∞;

NN-Index-Simple-TS(pa, q)          // pa = Diskadress z.B. d
    result = ∅;
    p := pa.loadPage();
    IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
                result := getObject(i);
                stopdist = dist(q, p.getObject(i));
    ELSE                             // p ist Directoryseite
        FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ stopdist THEN
                result := NN-Index-Simple-TS(p.childPage(i), q)
    RETURN result;
```

Objekt o3 in p22

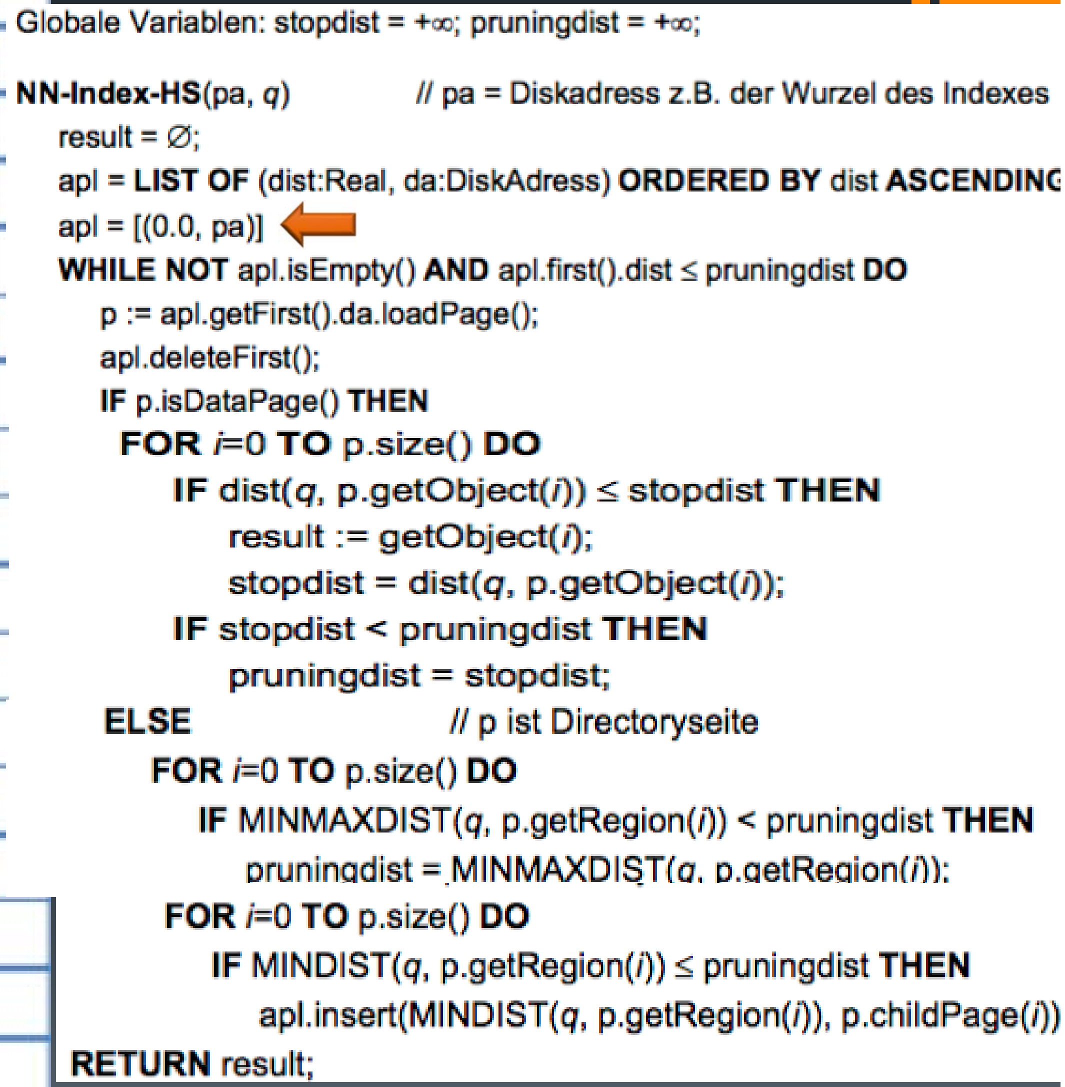
dist(Q,o3) = 6.5 <= 4?
Nein → true drop

A5-2 (b)

Besuchte Seiten:



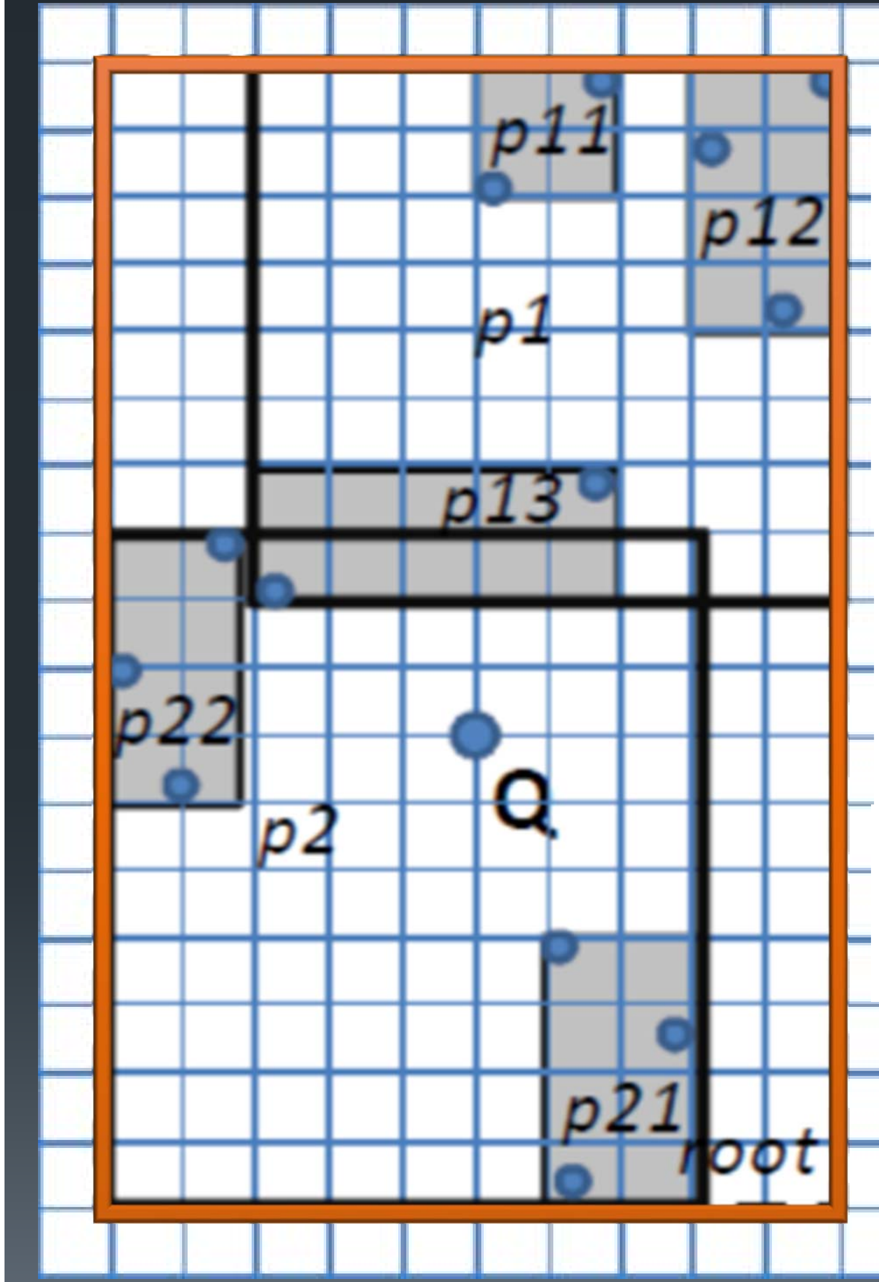
stopdist = +Inf   pruningdist= +Inf



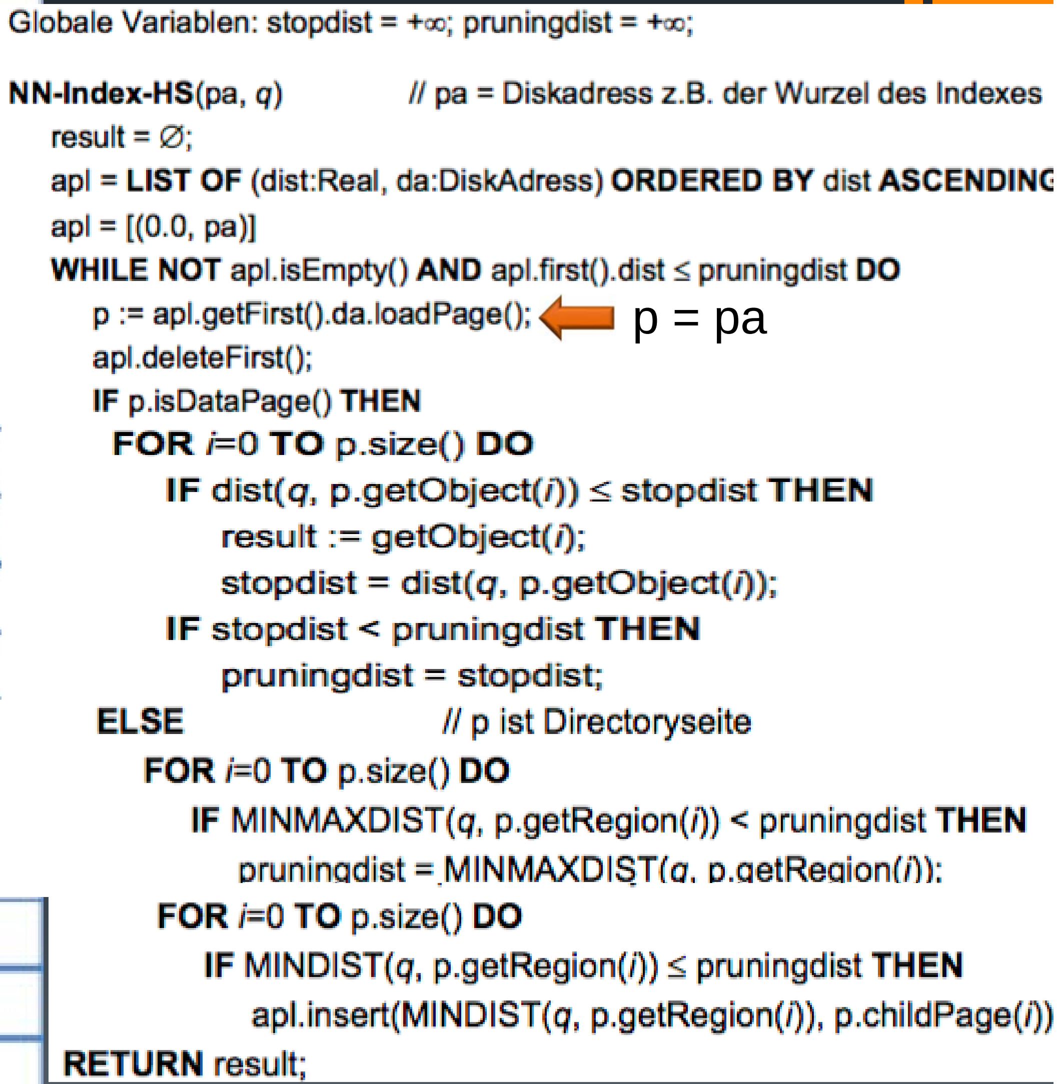
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
       IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
       IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(0,pa)]

Besuchte Seiten:



stopdist = +Inf pruningdist= +Inf

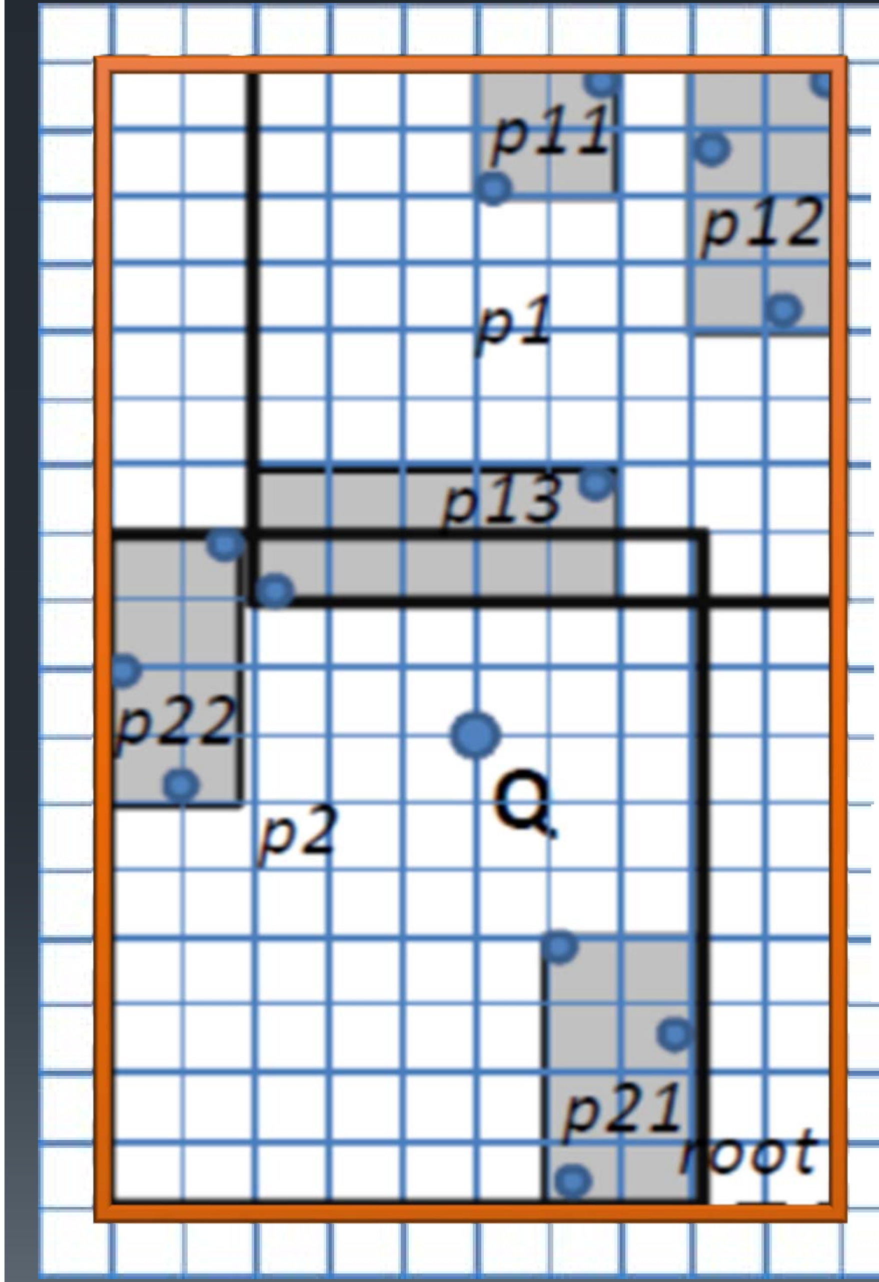


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
           IF dist(q, p.getObject(i)) ≤ stopdist THEN
              result := getObject(i);
              stopdist = dist(q, p.getObject(i));
           IF stopdist < pruningdist THEN
              pruningdist = stopdist;
      ELSE                          // p ist Directoryseite
         FOR i=0 TO p.size() DO
            IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
               pruningdist = MINMAXDIST(q, p.getRegion(i));
         FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
               apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

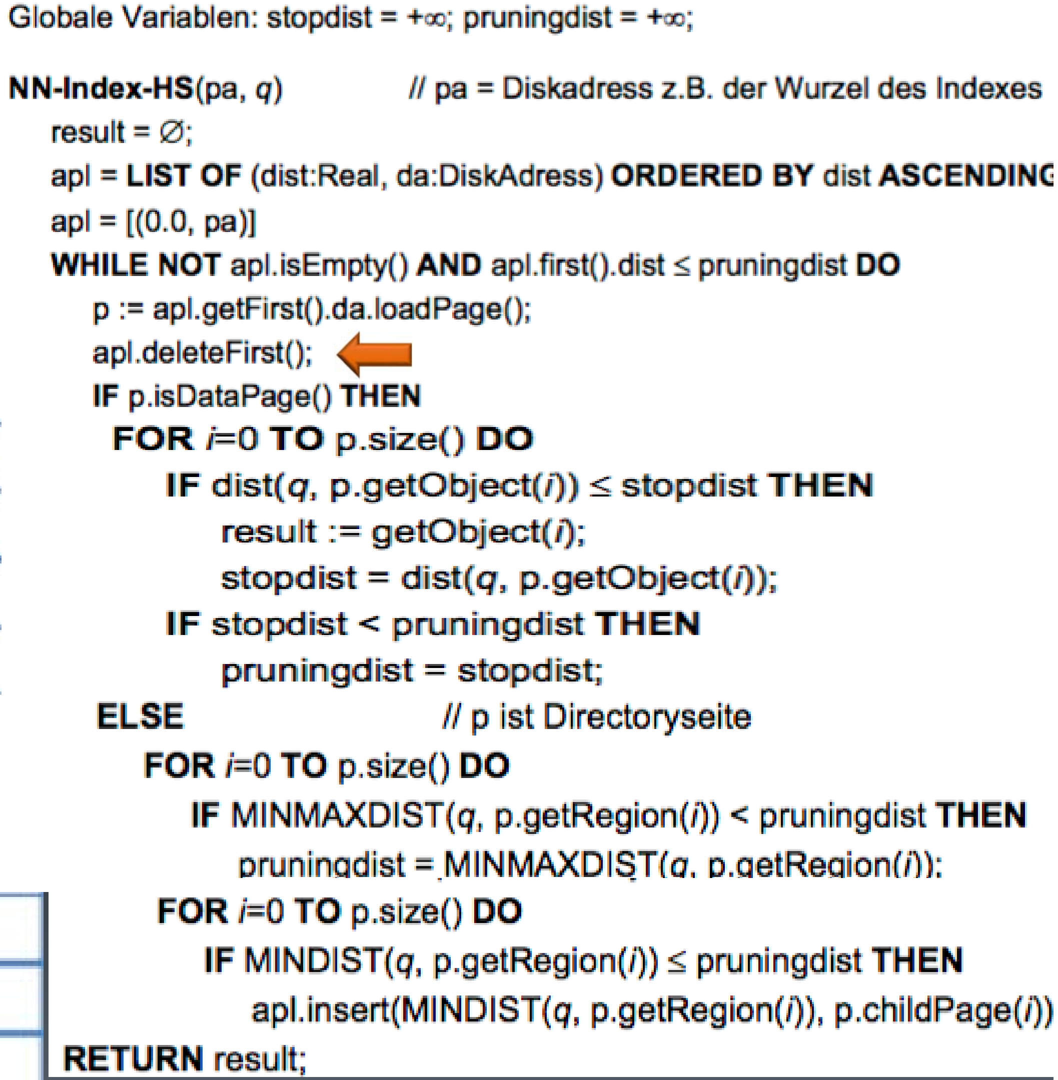
p = pa

APL= [(0,pa)]

Besuchte Seiten:



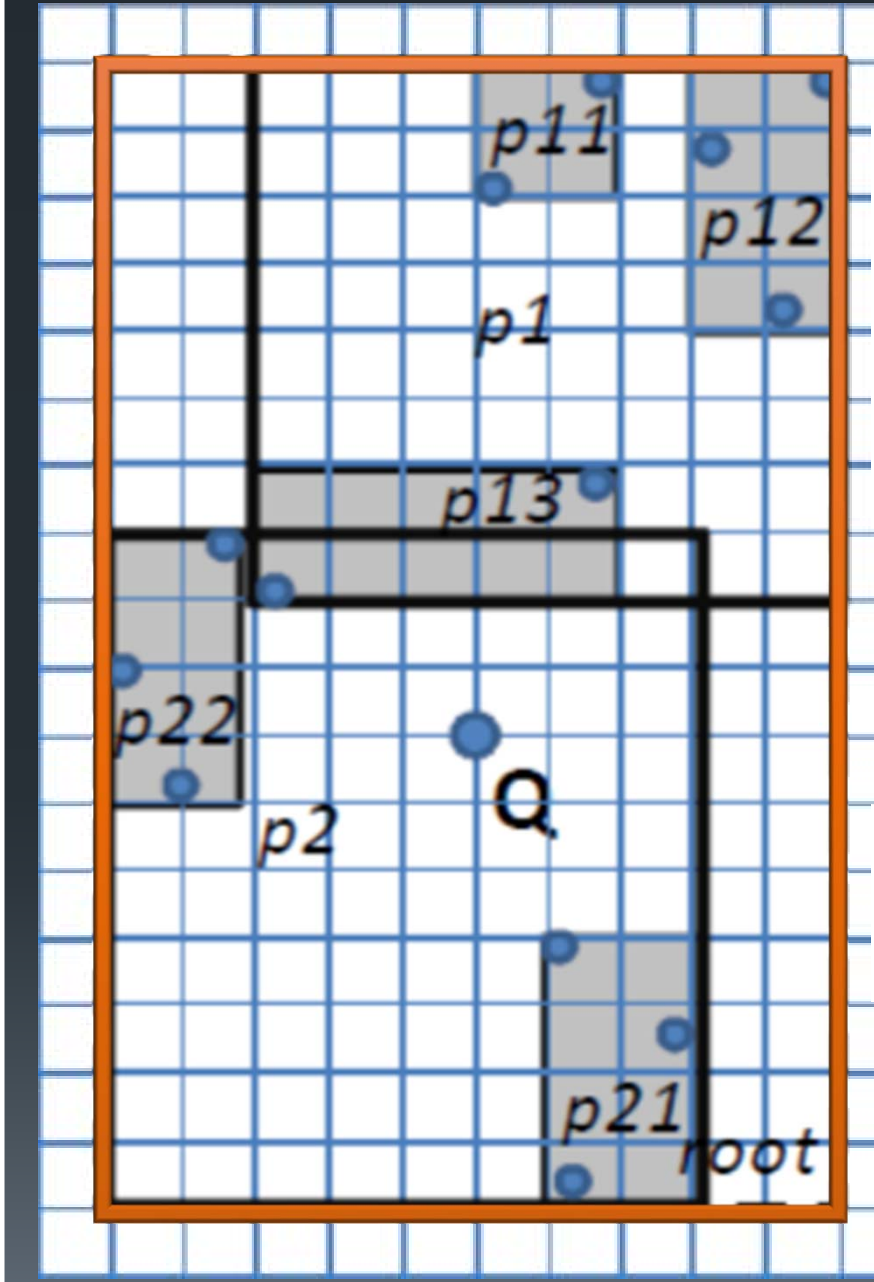
stopdist = +Inf pruningdist= +Inf



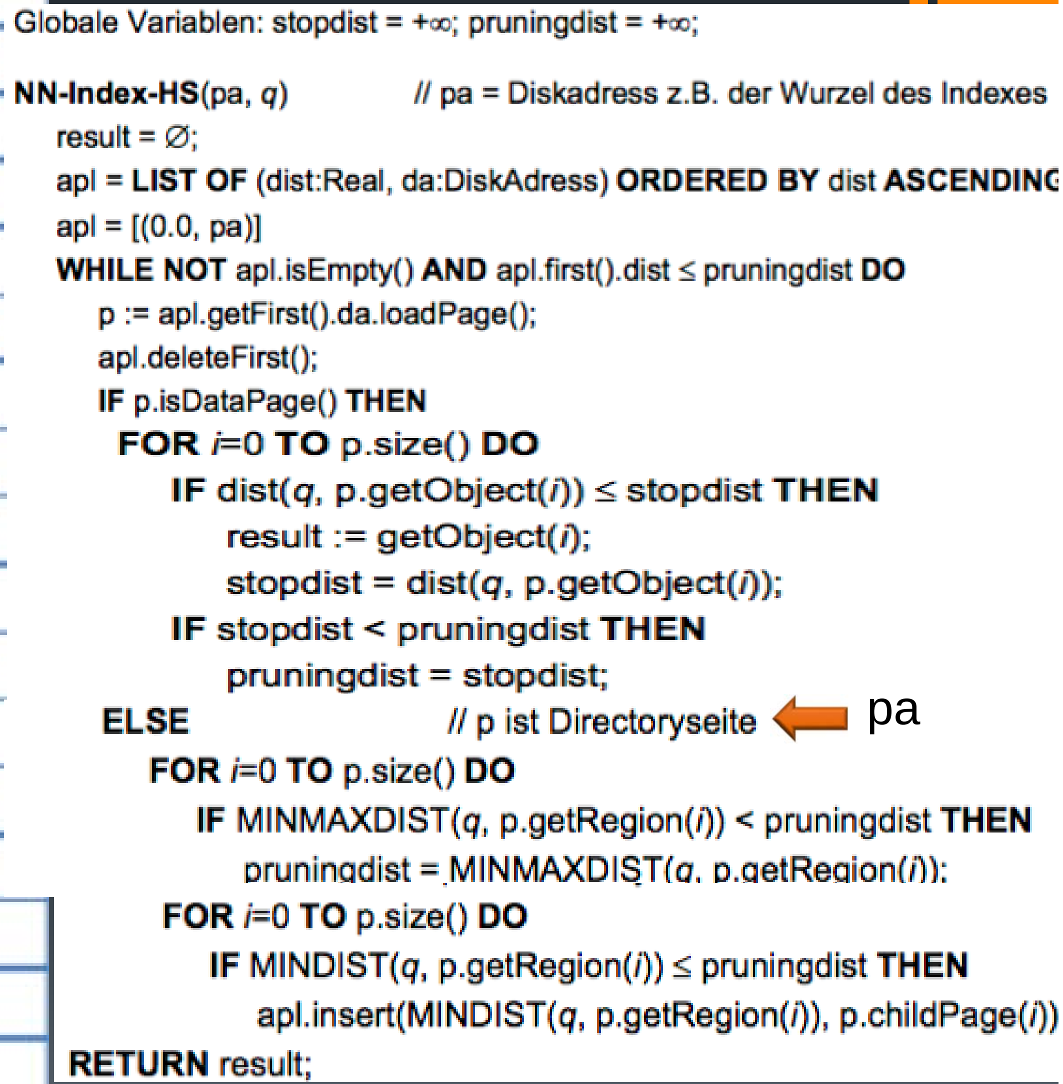
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= []

Besuchte Seiten:



stopdist = +Inf pruningdist= +Inf



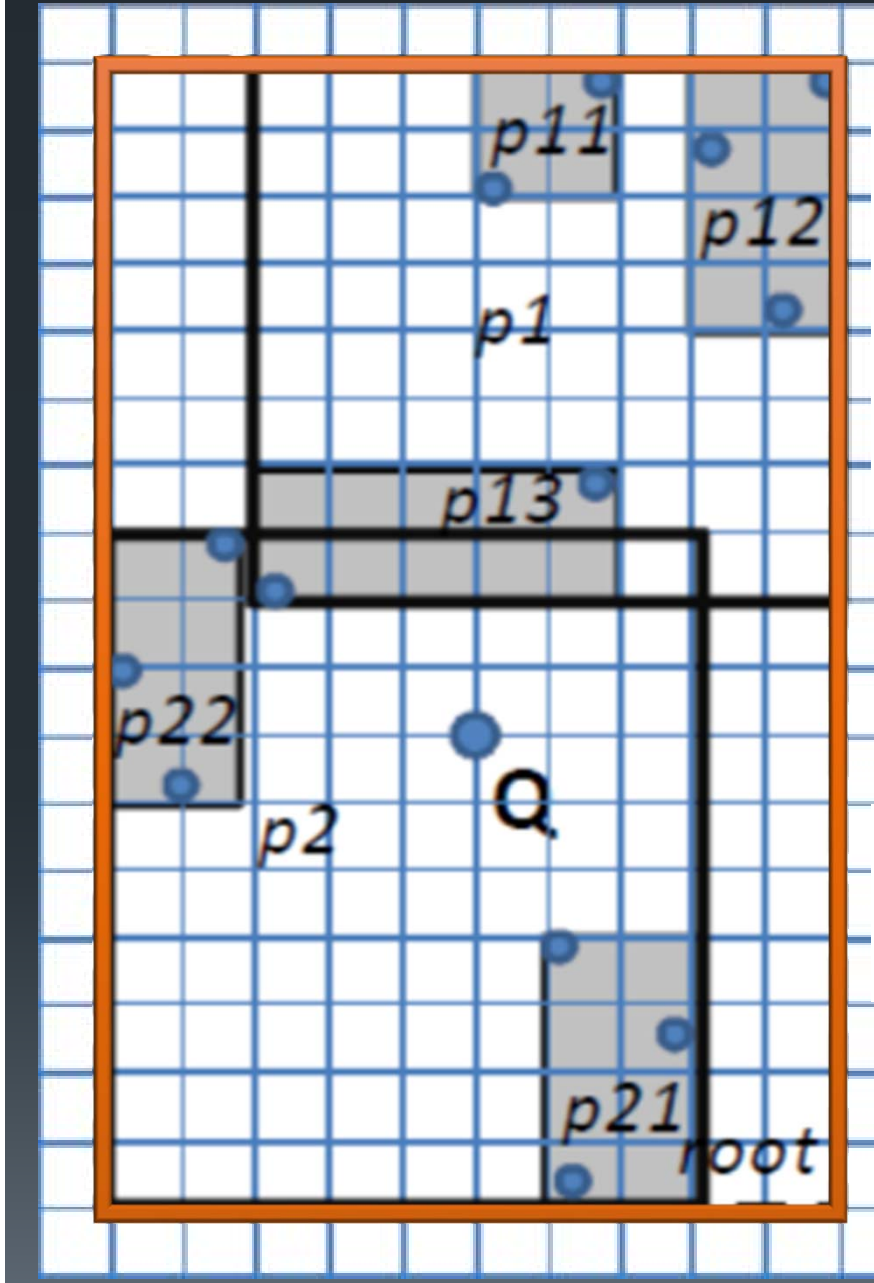
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                      // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

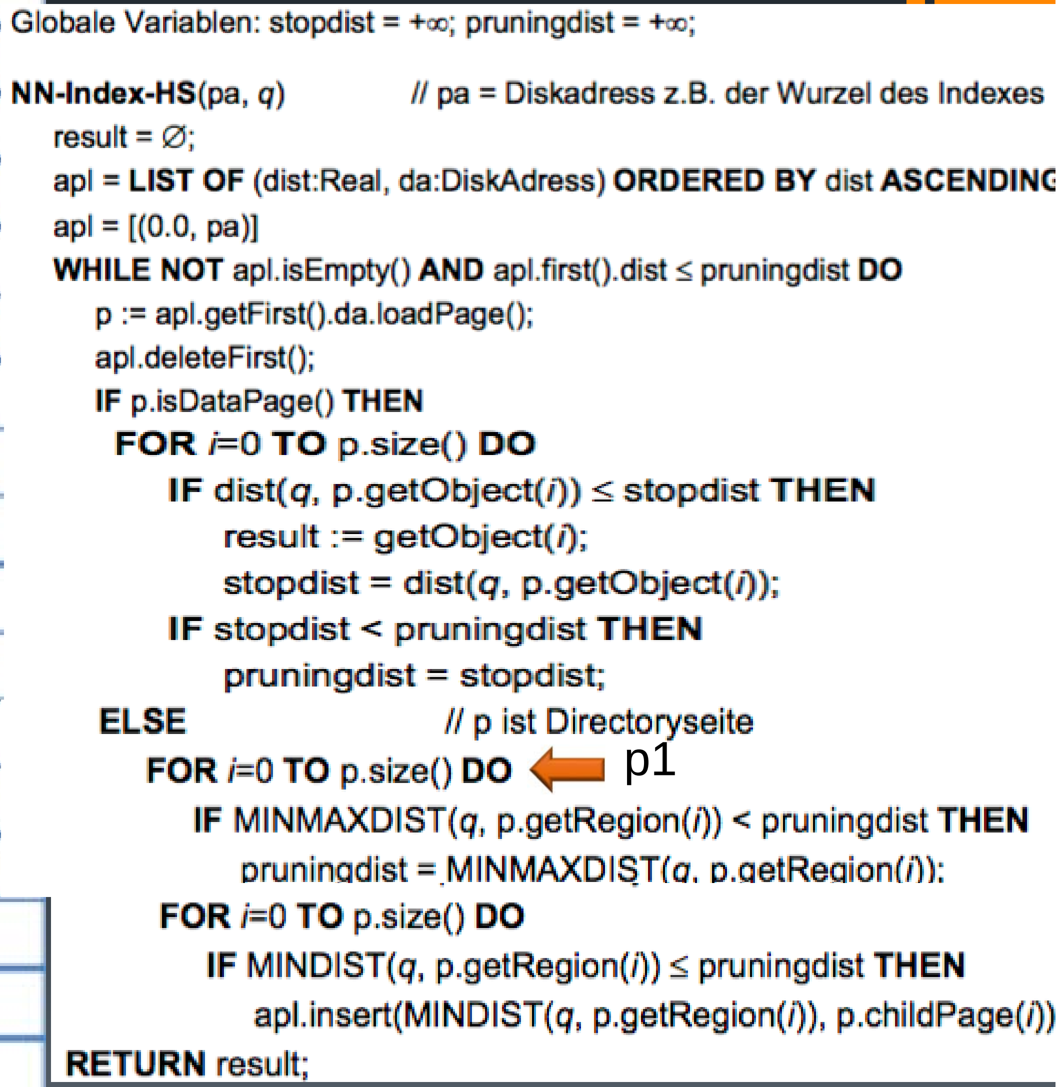
pa

APL= []

Besuchte Seiten:

stopdist = +Inf pruningdist= +Inf



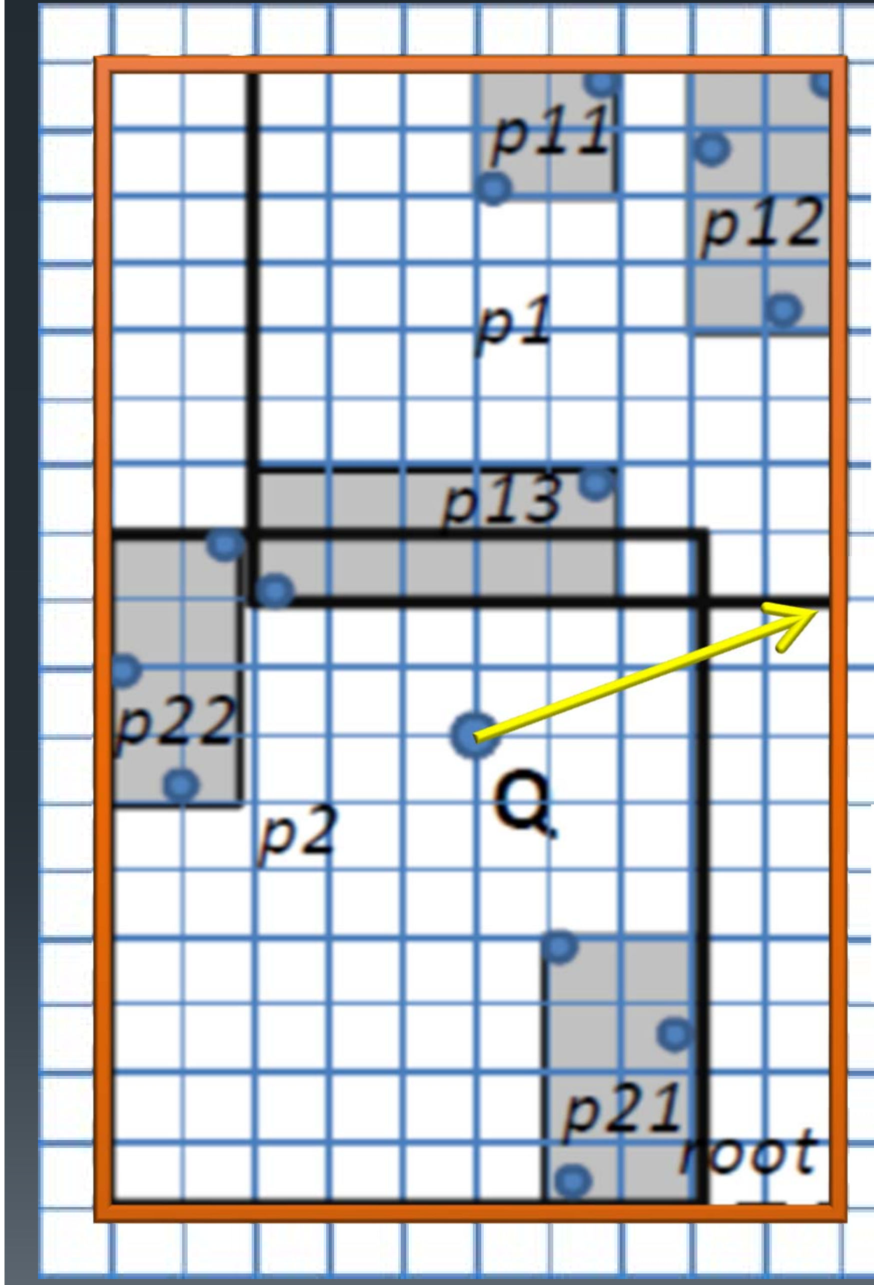


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

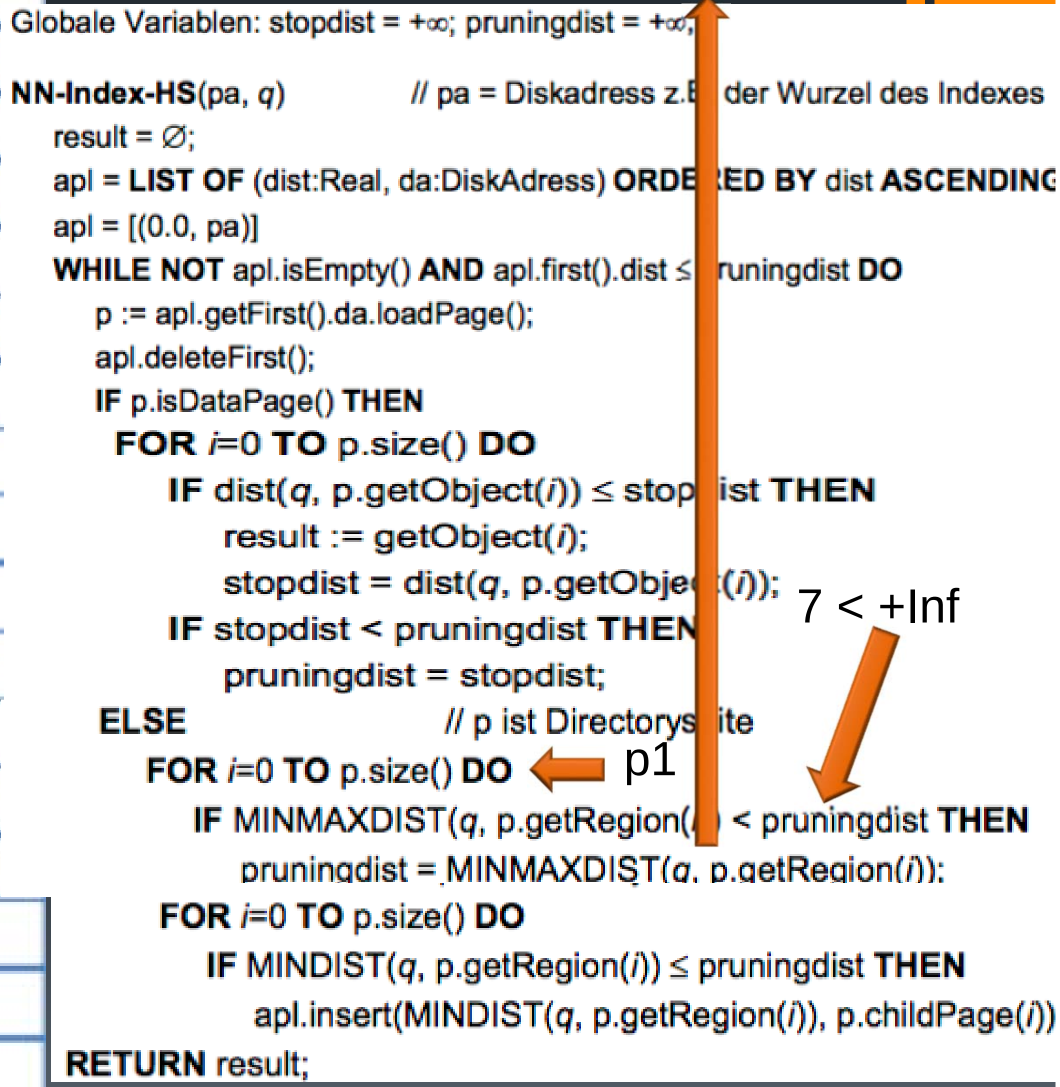
p1

APL= []

Besuchte Seiten:



stopdist = +Inf pruningdist= 7



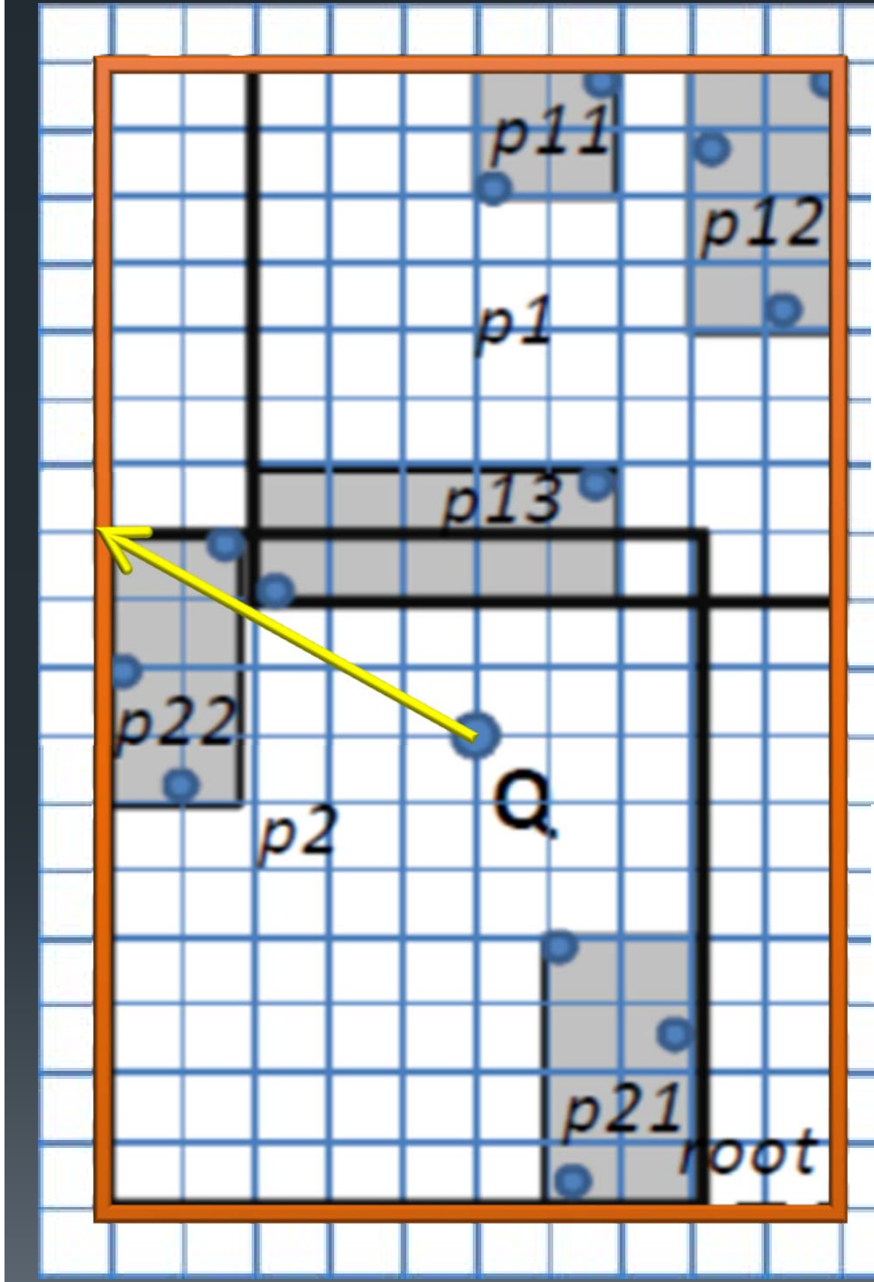
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.E der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDE ED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ runingdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
       IF dist(q, p.getObject(i)) ≤ stop ist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObje (i));
       IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directorys ite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion( ) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

7 < +Inf

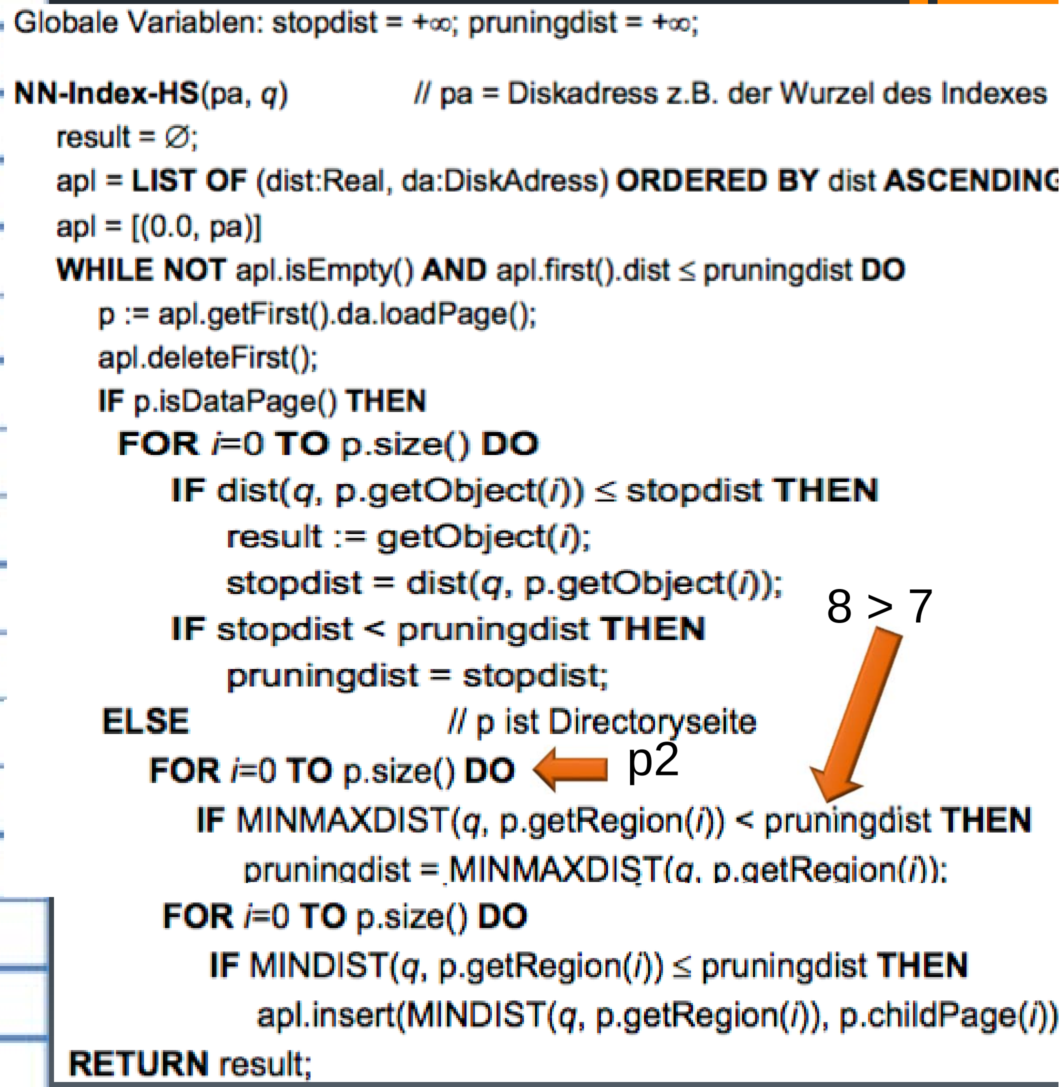
p1

APL= []

Besuchte Seiten:



stopdist = +Inf pruningdist= 7



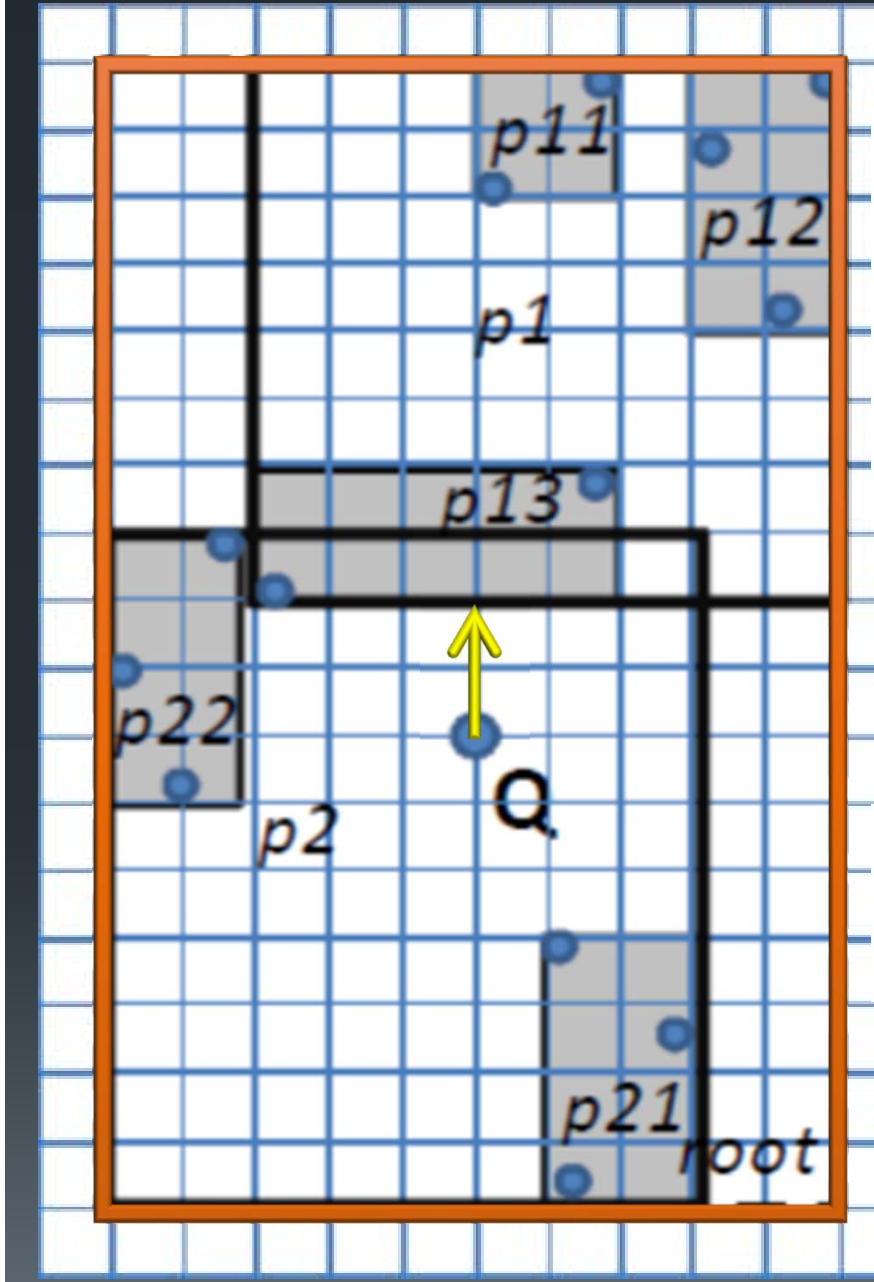
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
       IF dist(q, p.getObject(i)) ≤ stopdist THEN
         result := getObject(i);
         stopdist = dist(q, p.getObject(i));
       IF stopdist < pruningdist THEN
         pruningdist = stopdist;
    ELSE                    // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

8 > 7

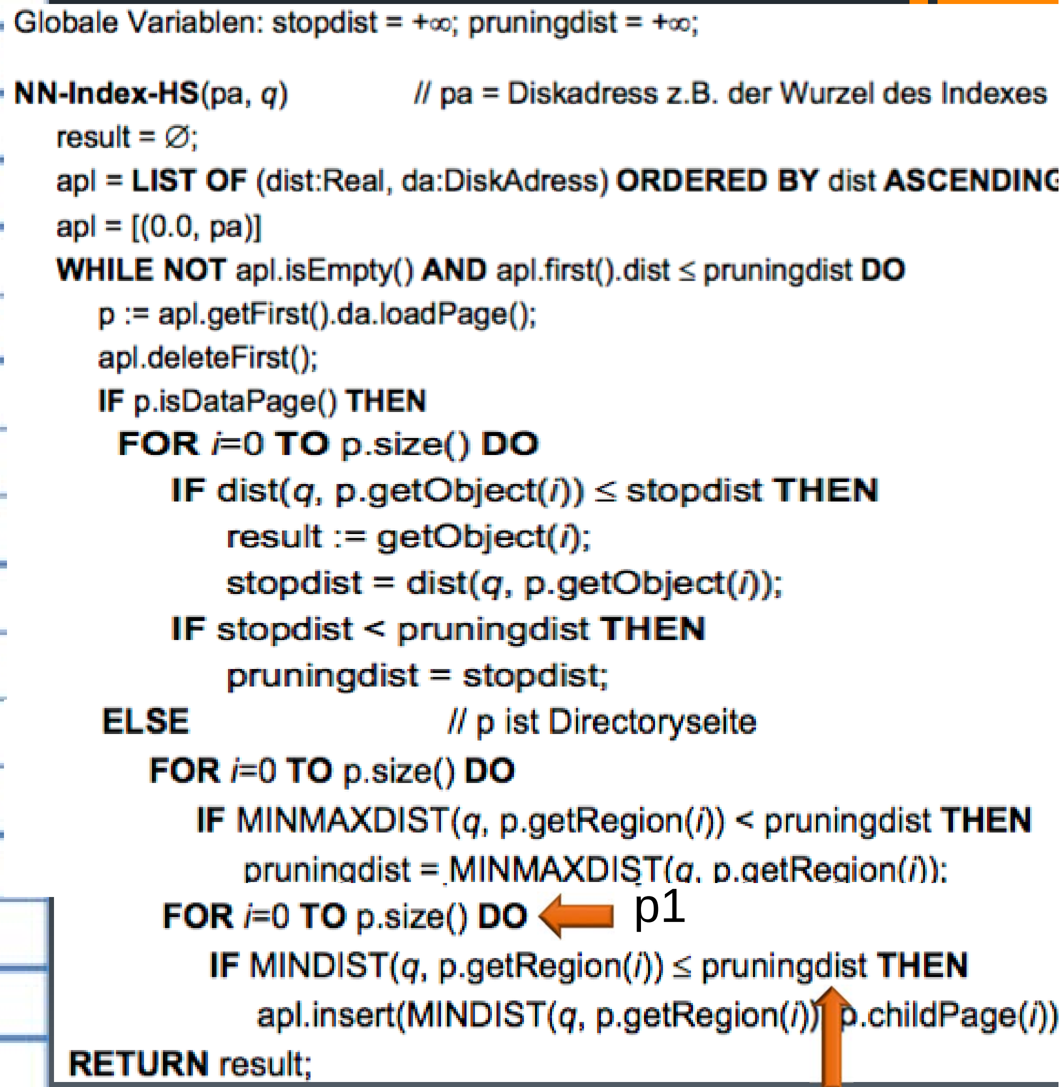
p2

APL= []

Besuchte Seiten:



stopdist = +Inf   pruningdist= 7

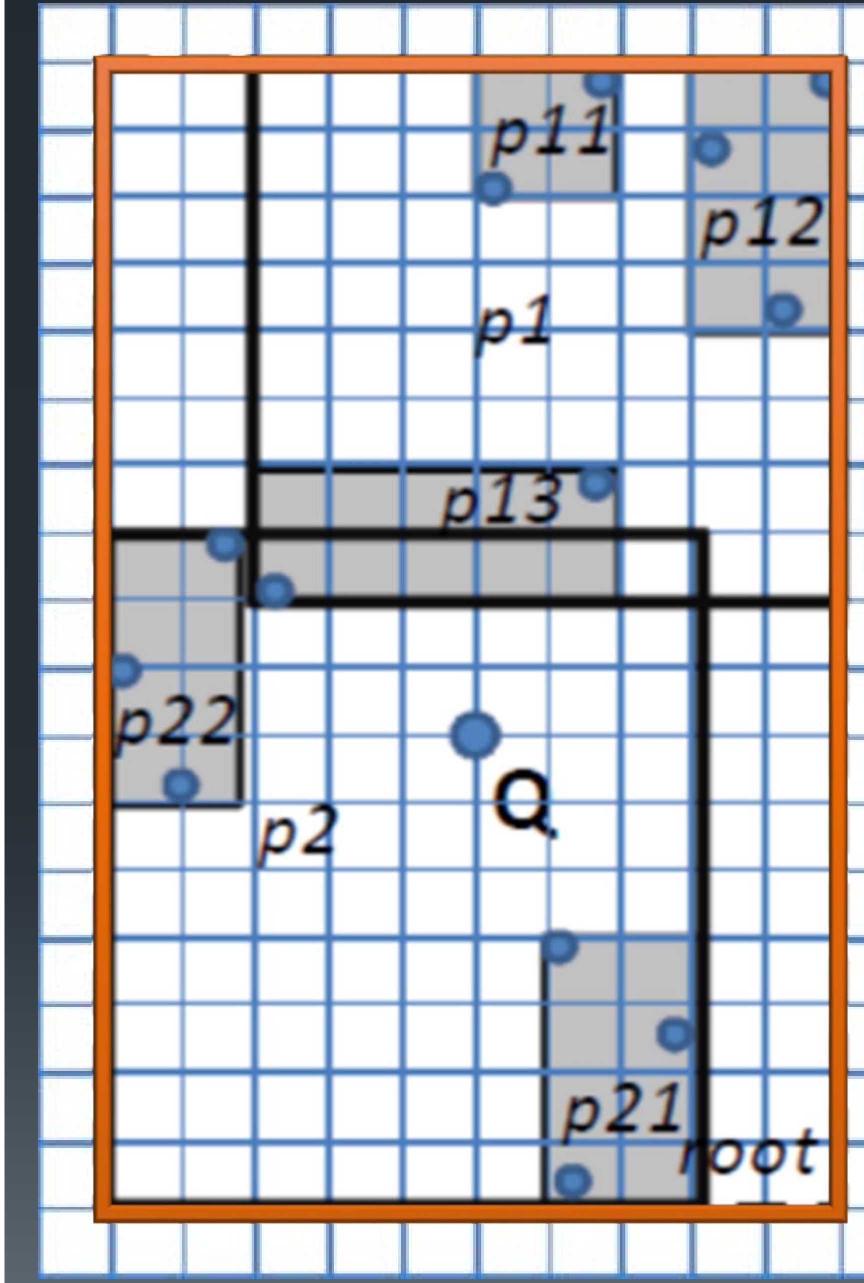


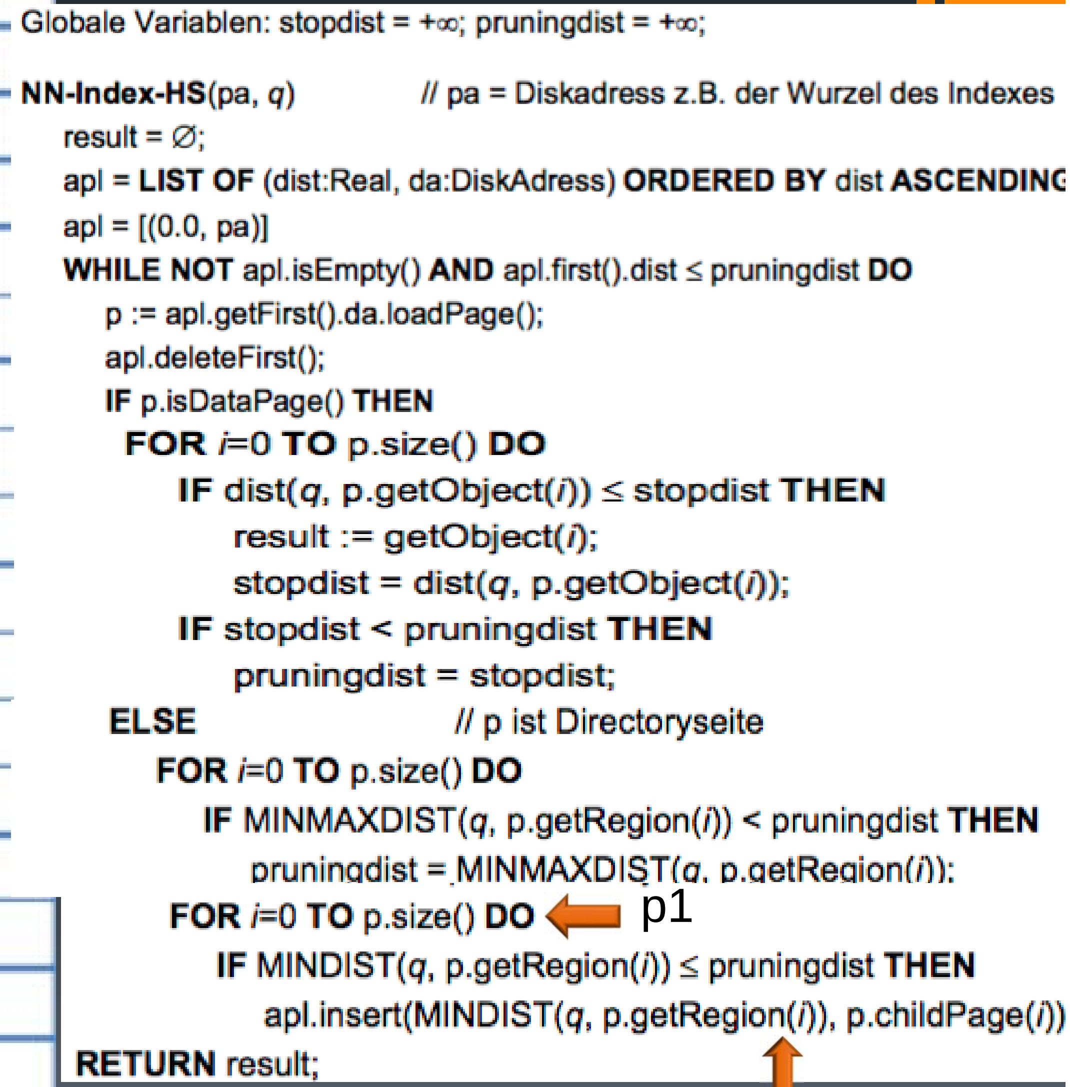
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO  p1
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= []   MINDIST(q,p1)= 2 <= 7

Besuchte Seiten:

stopdist = +Inf pruningdist= 7

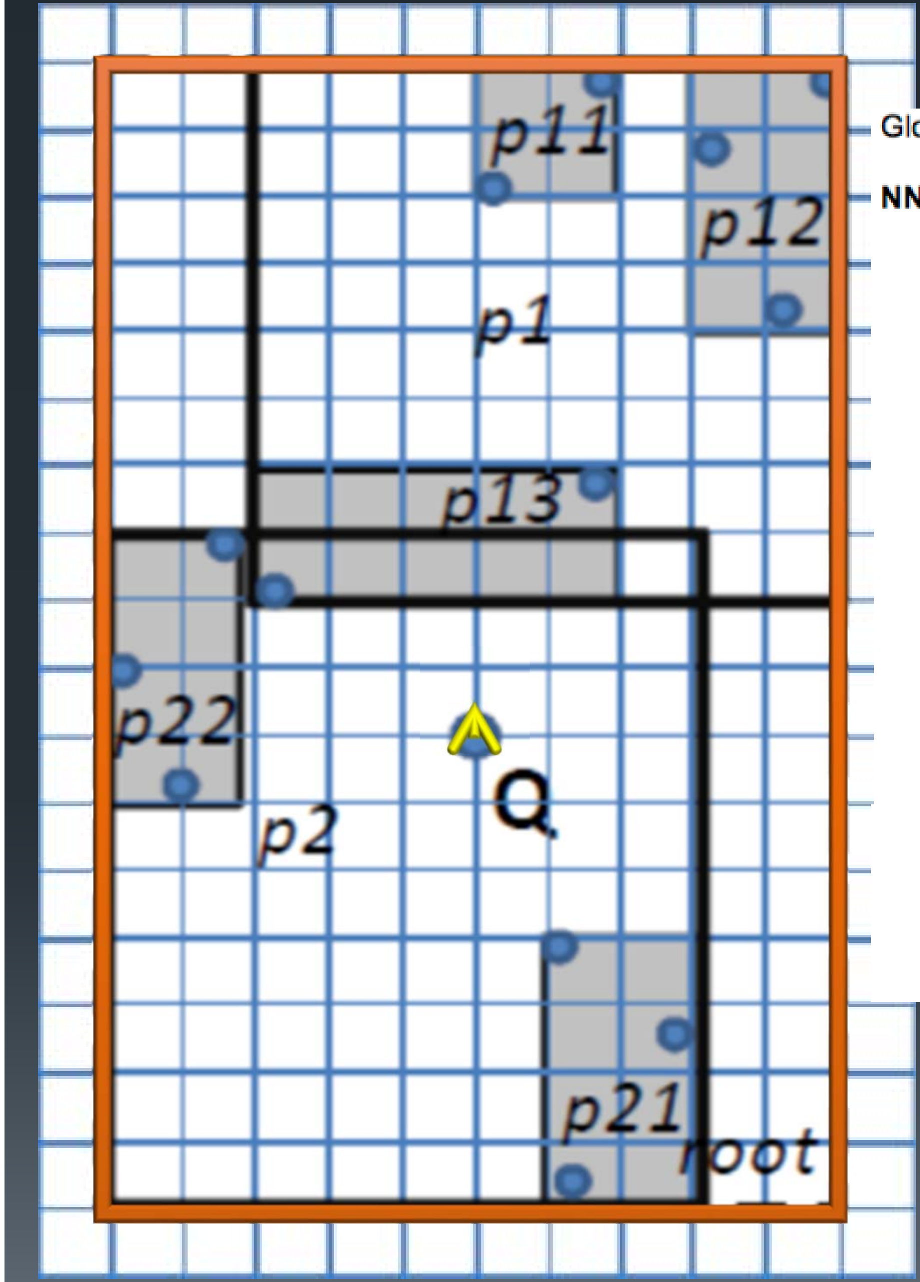




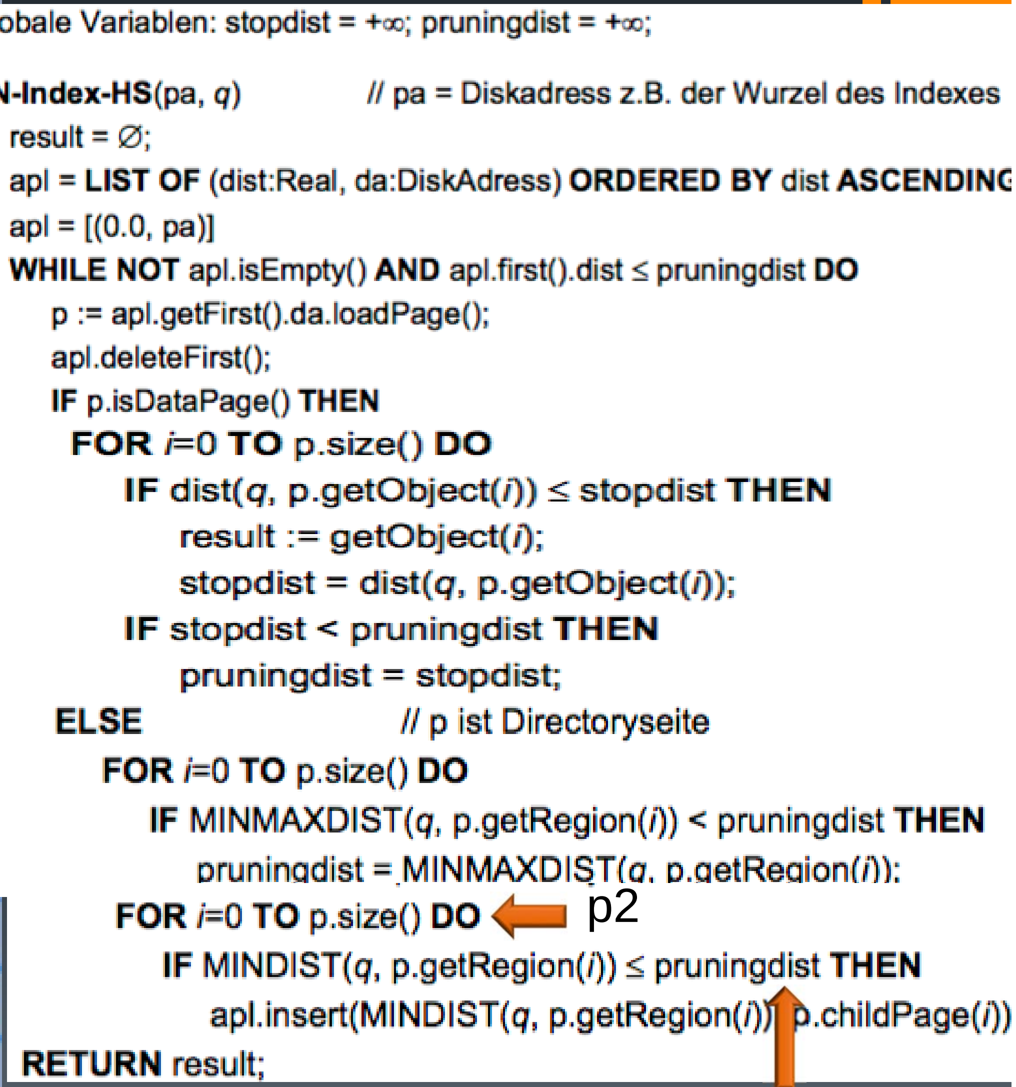
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                      // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO  p1
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(2,p1)]

Besuchte Seiten:



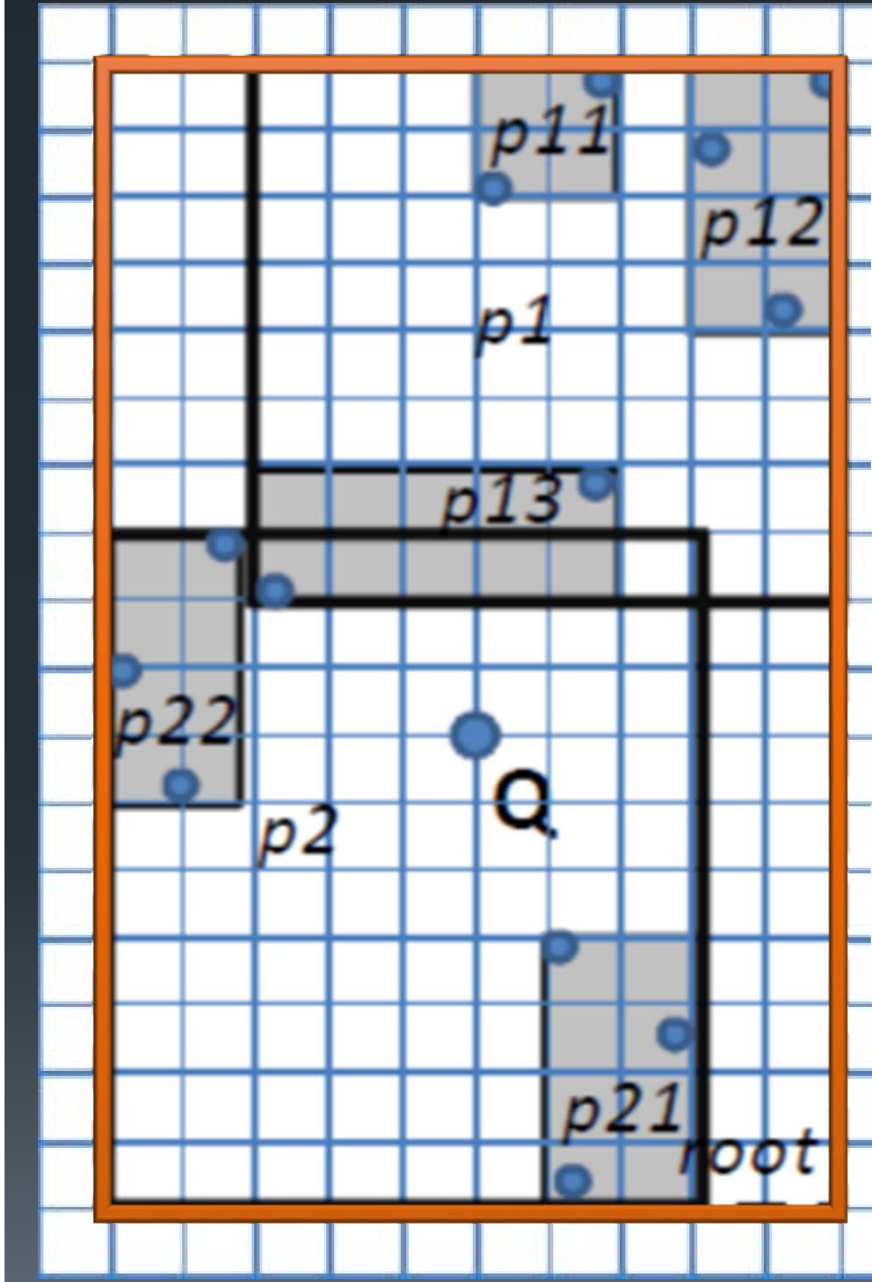
stopdist = +Inf   pruningdist= 7



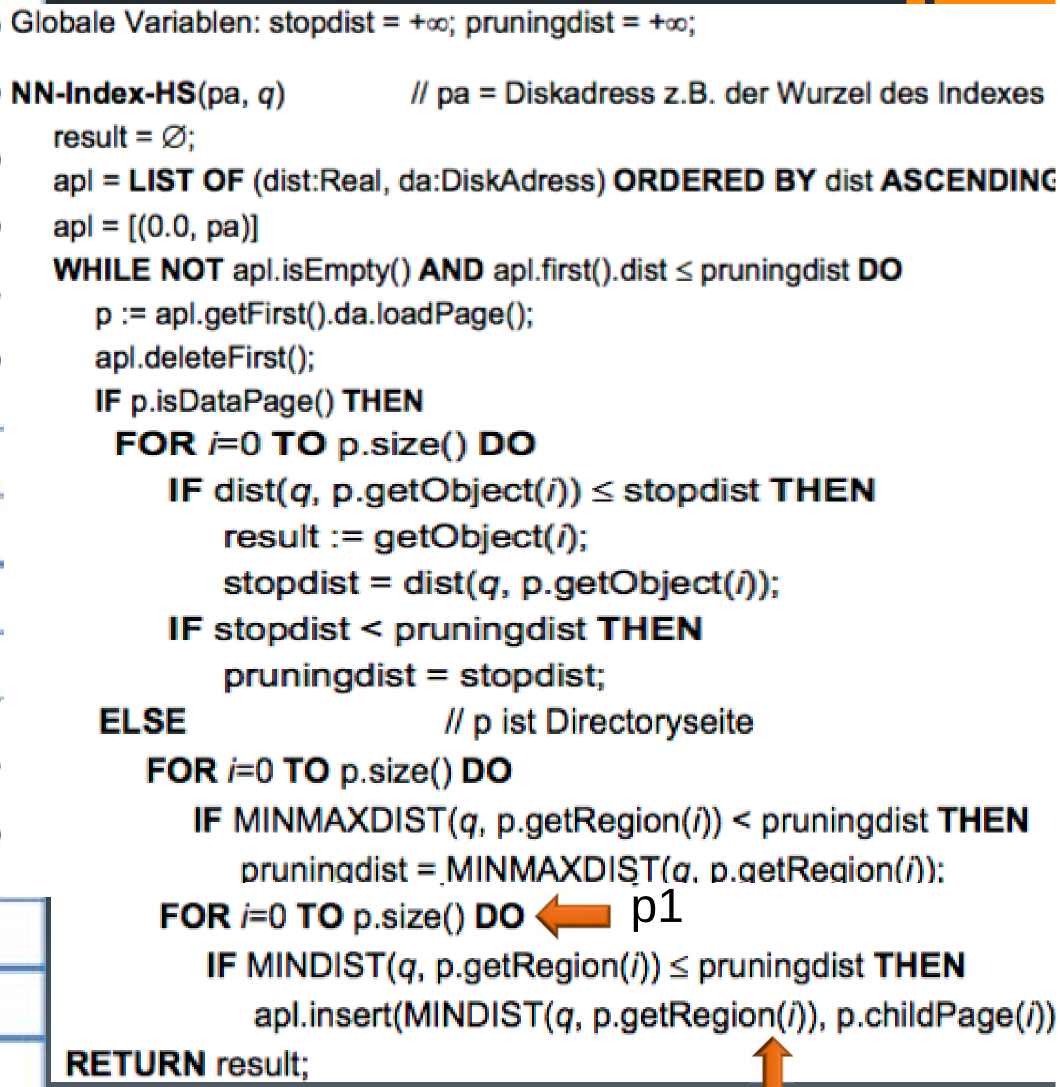
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                           // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO  ⇐ p2
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= [(2,p1)]   MINDIST(q,p2)= 0 <= 7

Besuchte Seiten:



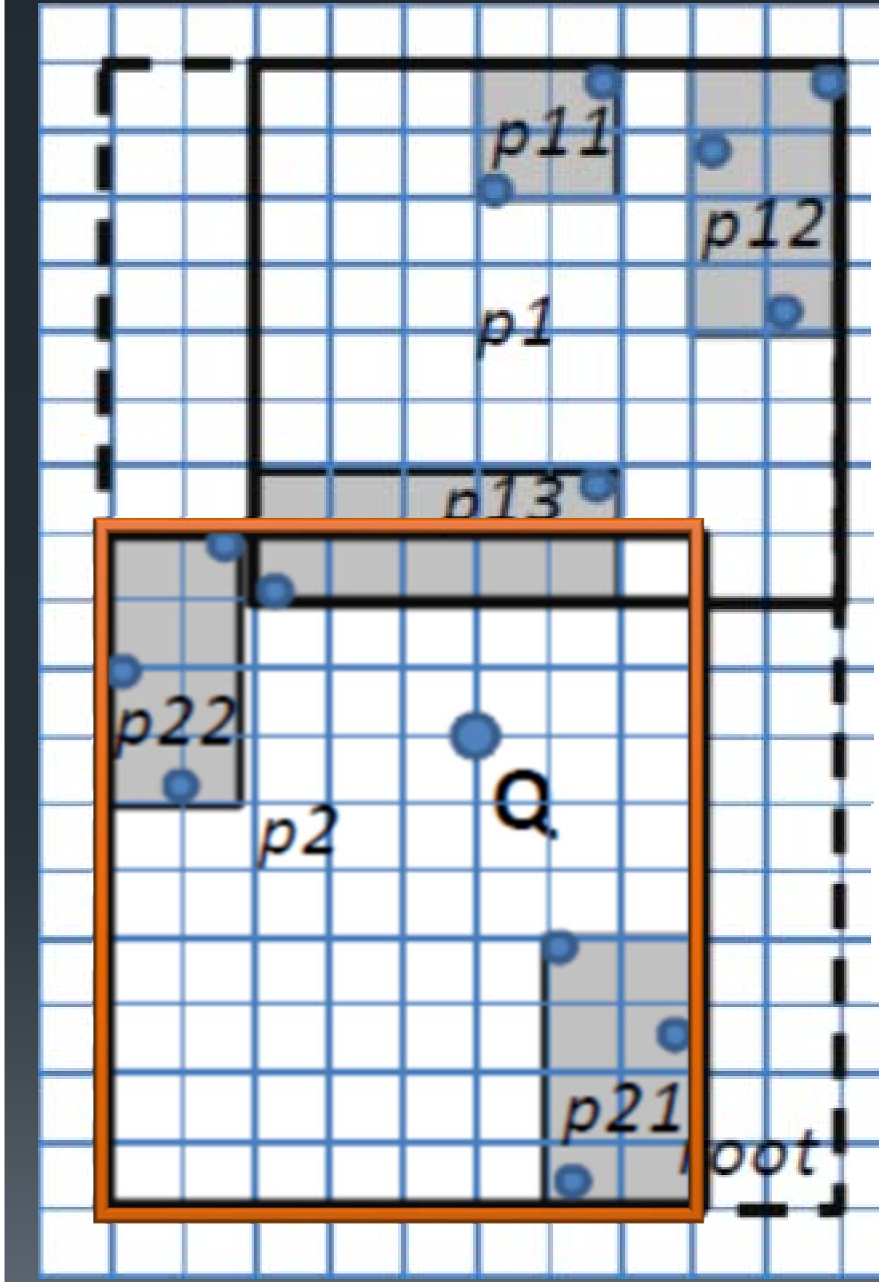
stopdist = +Inf   pruningdist= 7



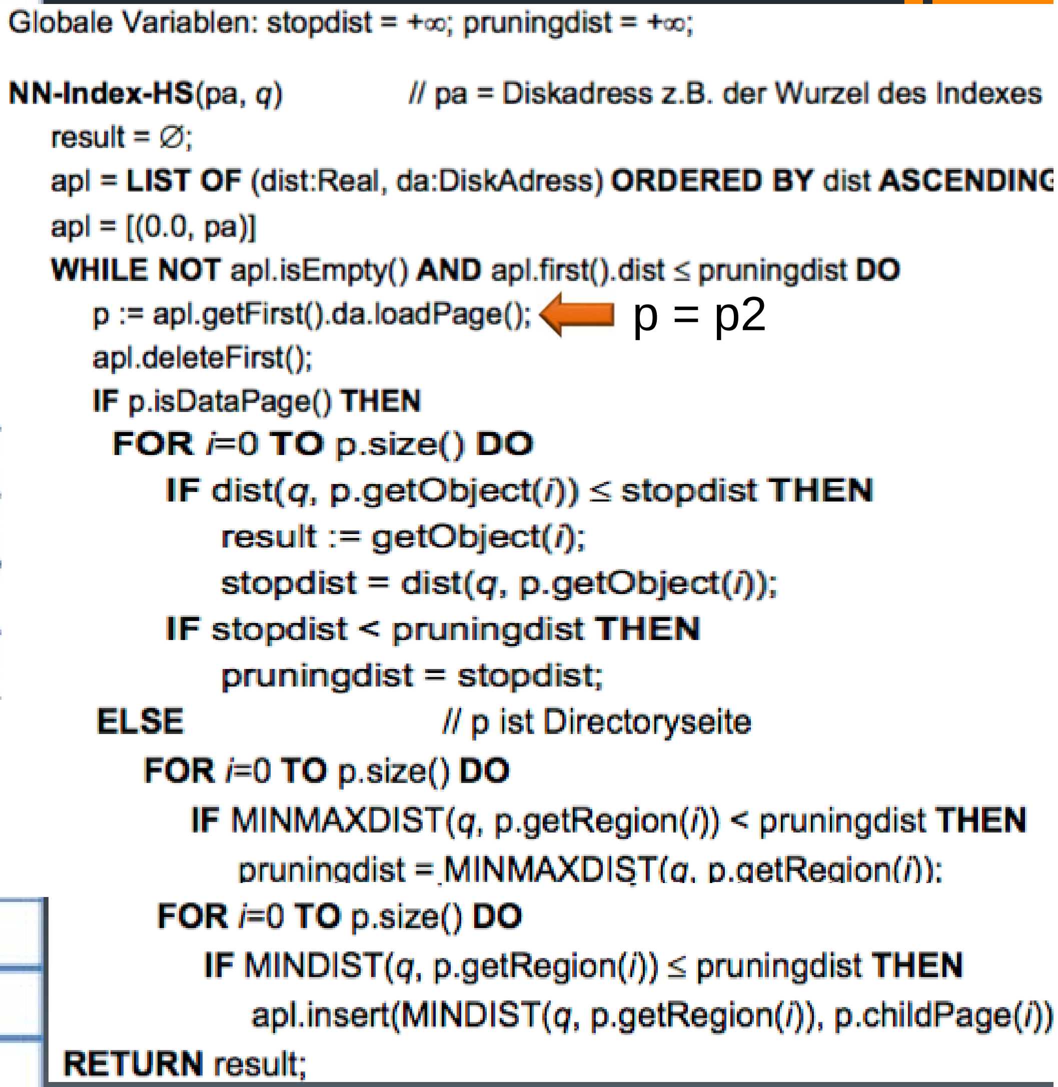
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO    p1
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(0,p2),(2,p1)]

Besuchte Seiten: p2



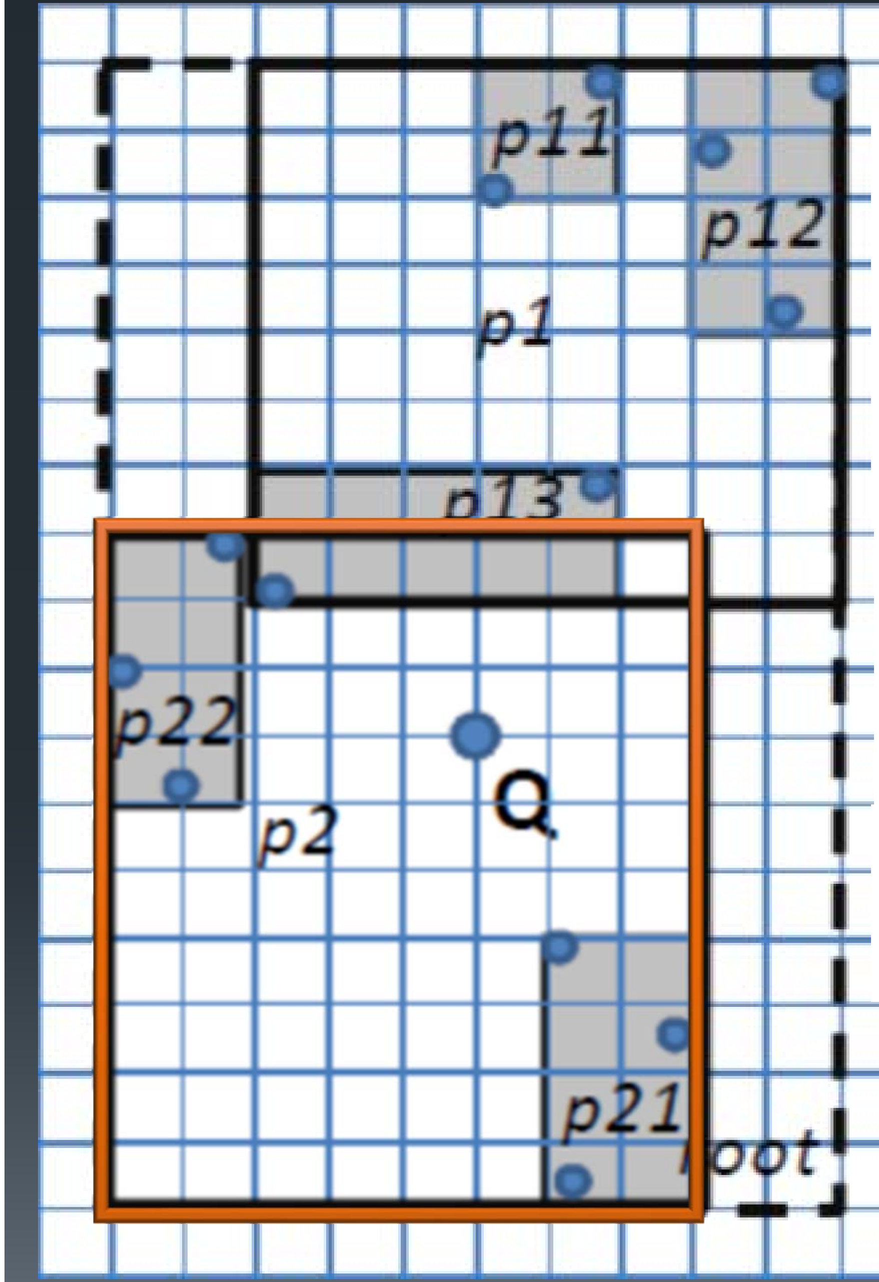
stopdist = +Inf pruningdist= 7



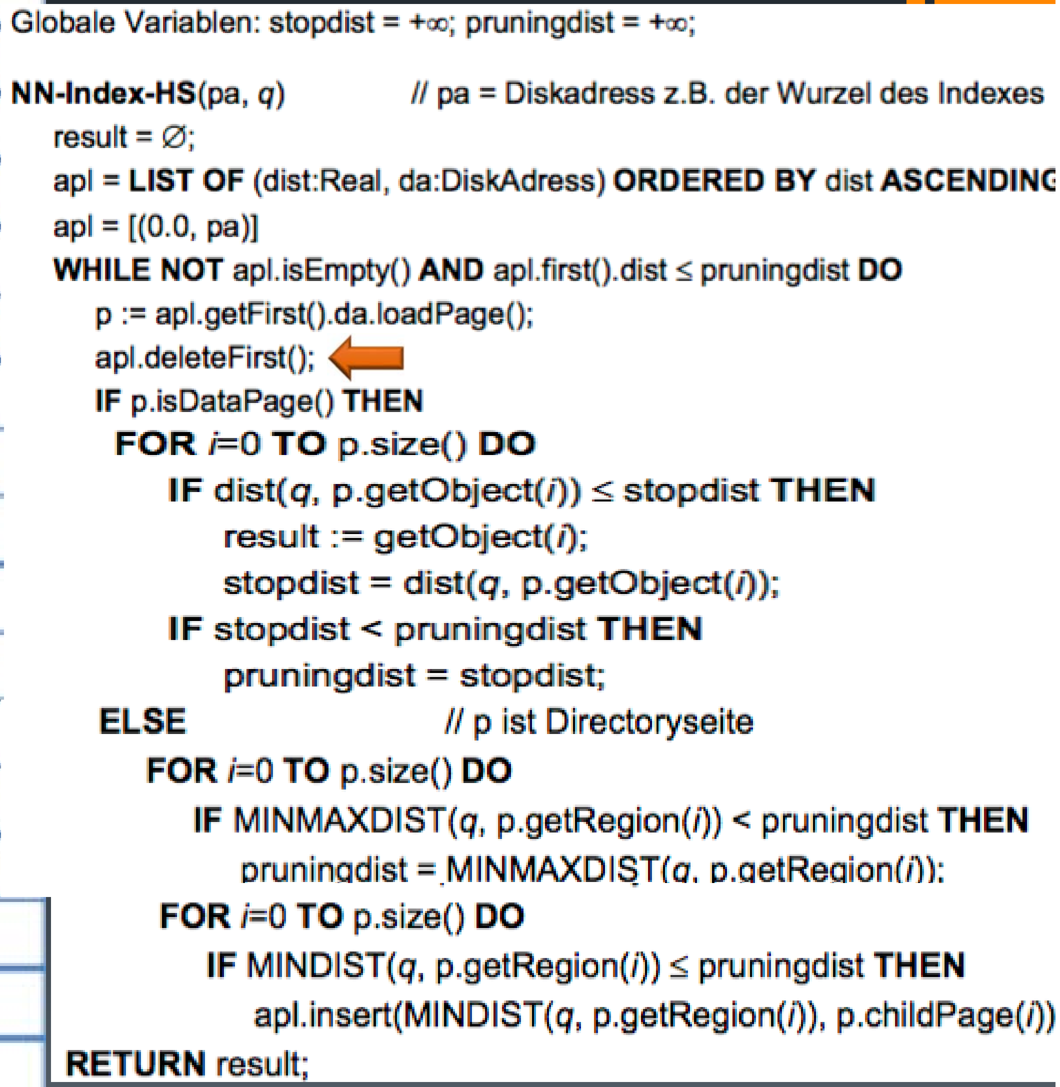
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();  ⇐  p = p2
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(0,p2),(2,p1)]

Besuchte Seiten: p2



stopdist = +Inf   pruningdist= 7

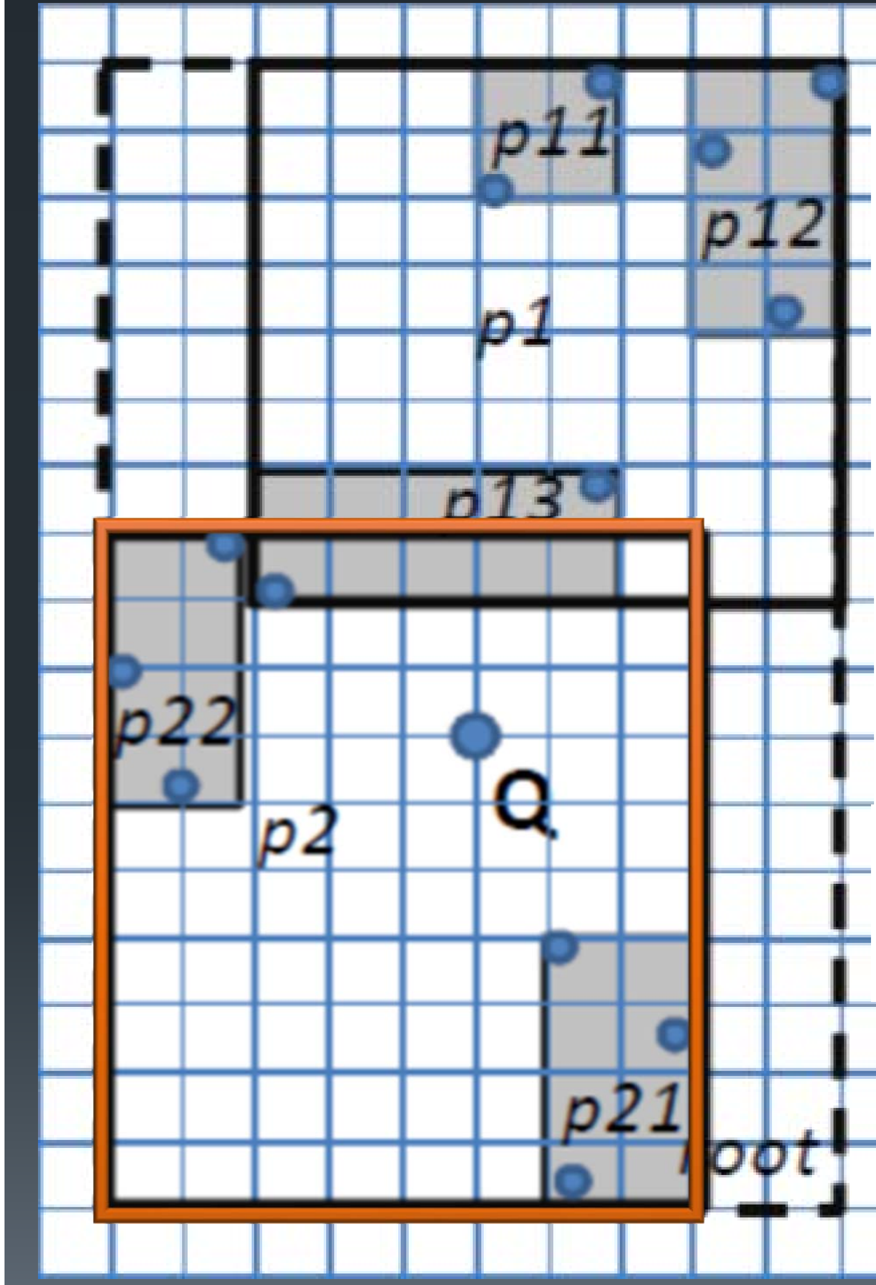


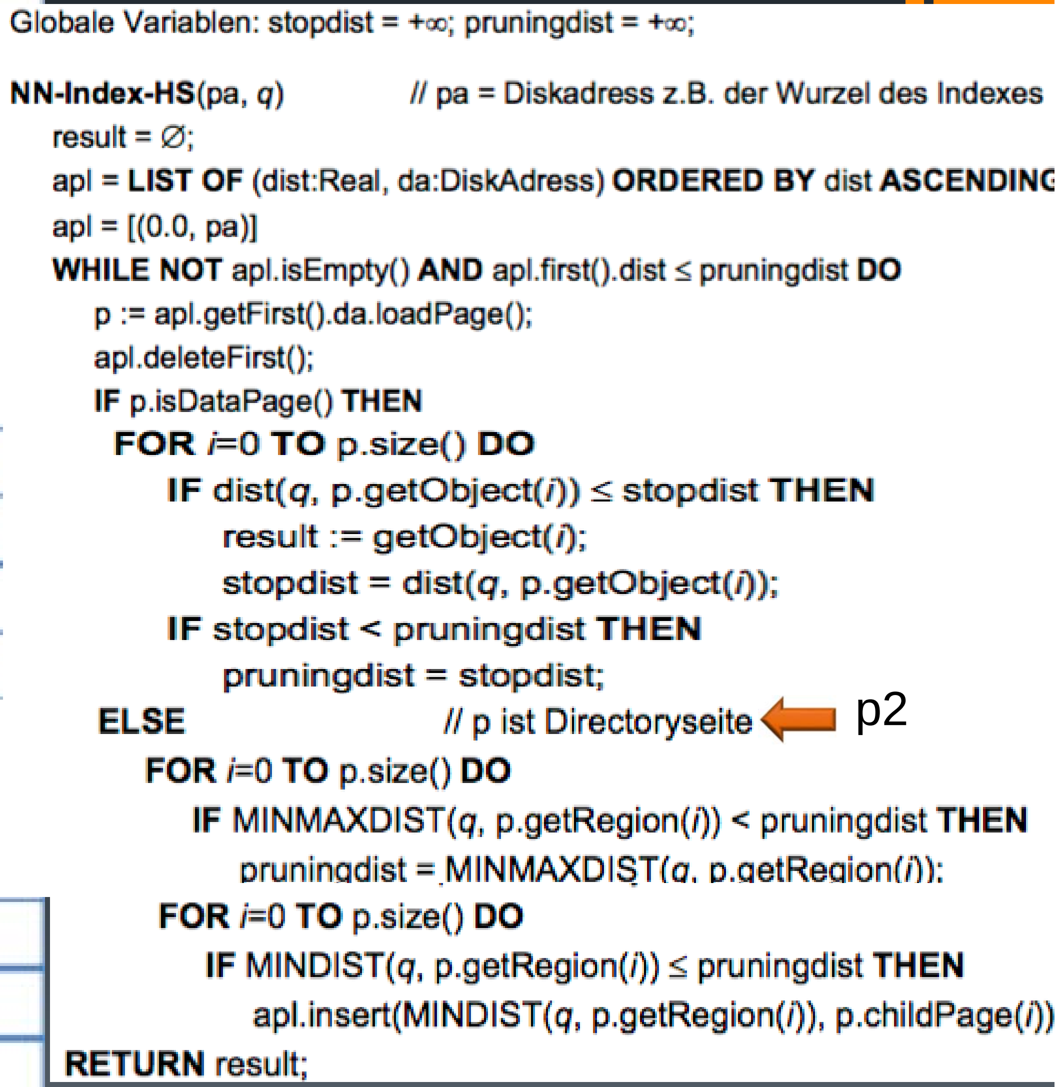
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                              // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= [(2,p1)]

Besuchte Seiten: p2

stopdist = +Inf pruningdist= 7





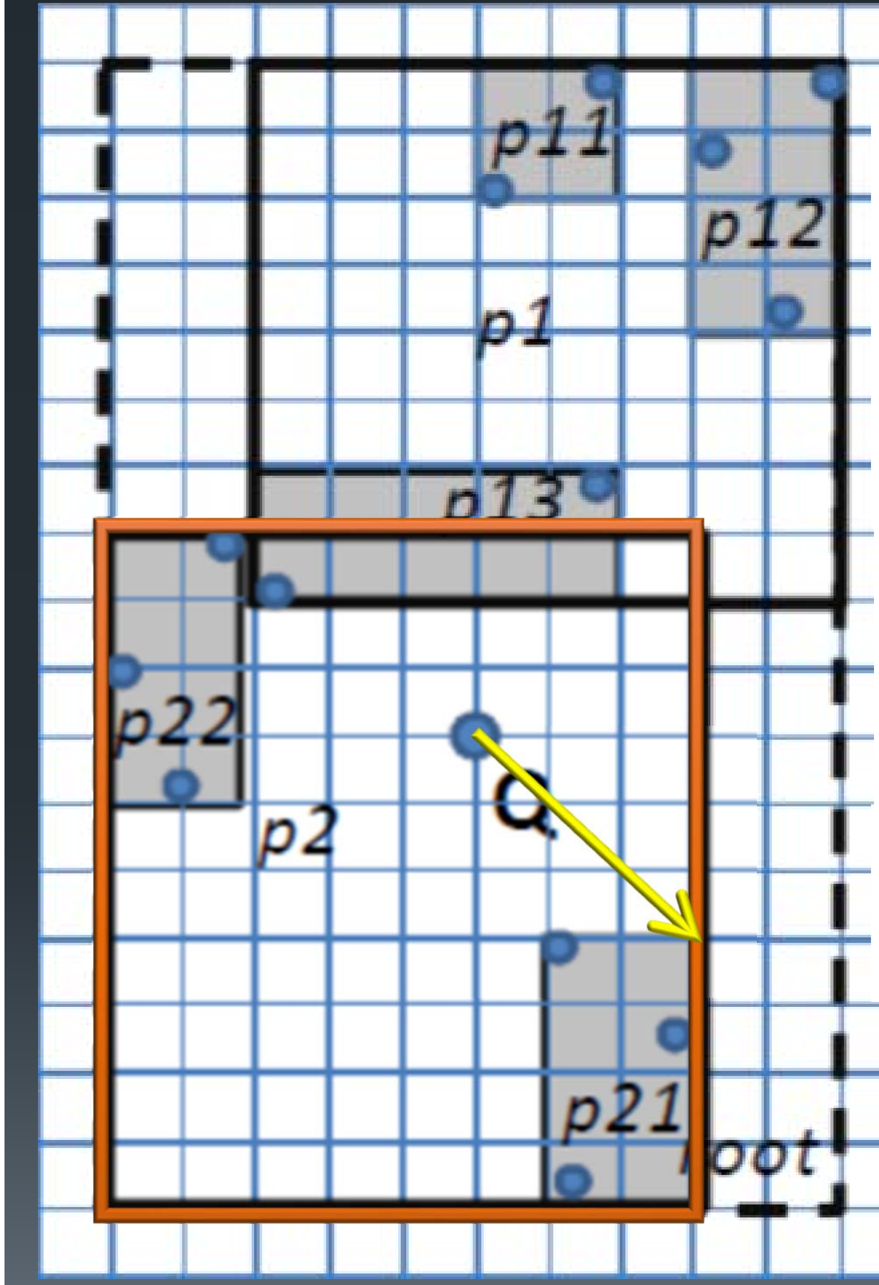
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                      // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

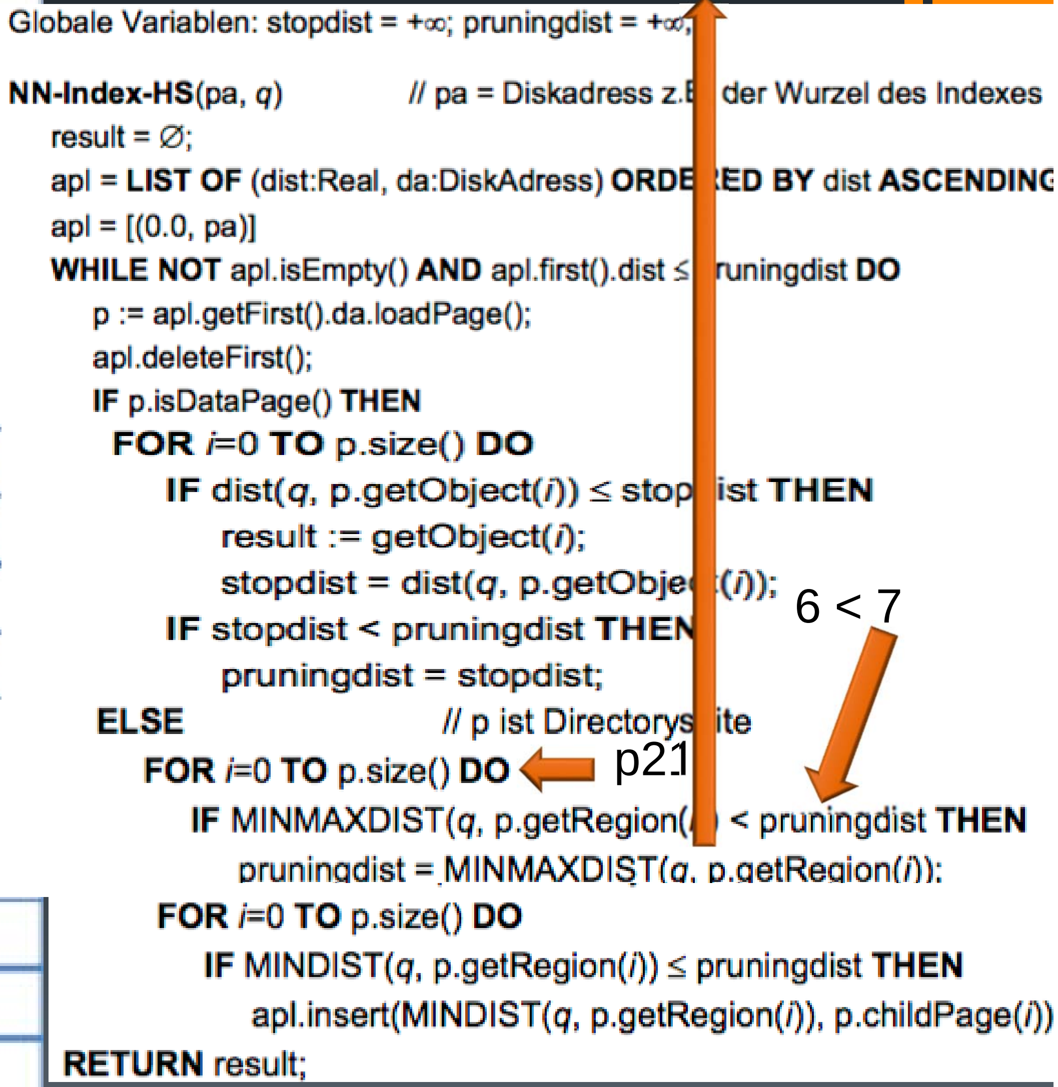
p2

APL= [(2,p1)]

Besuchte Seiten: p2

stopdist = +Inf pruningdist= 6





```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                         // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

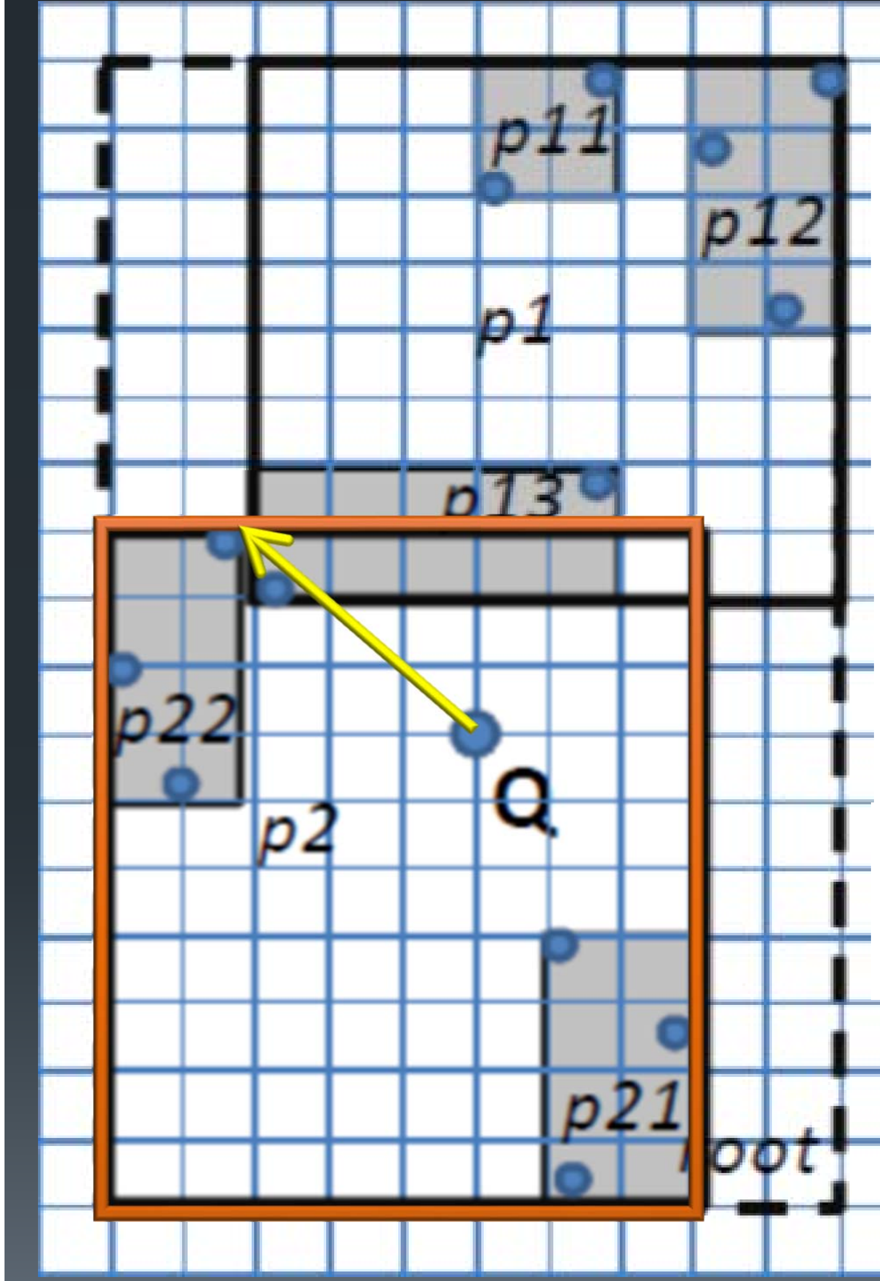
p21

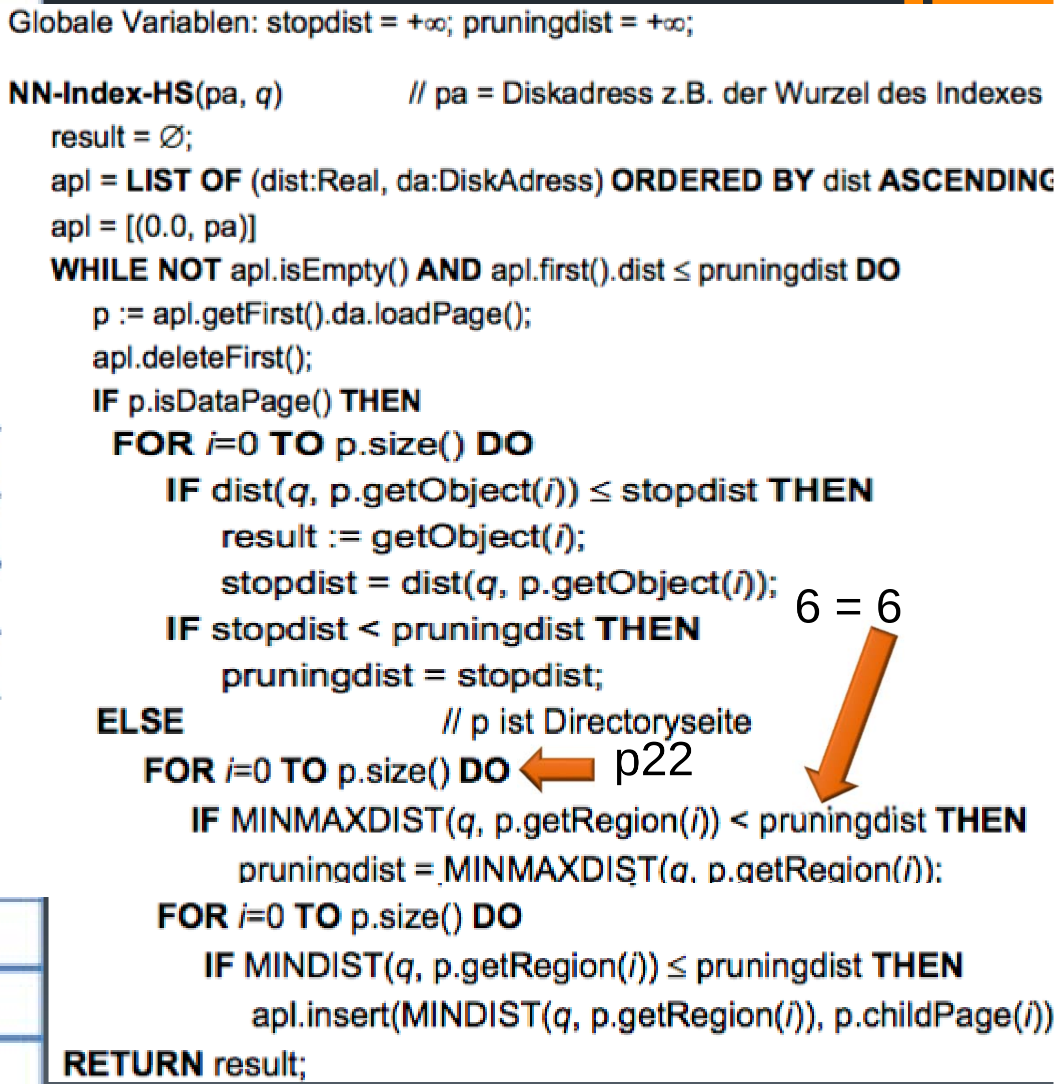
6 < 7

APL= [(2,p1)]

Besuchte Seiten: p2

stopdist = +Inf pruningdist= 6





```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

6 = 6

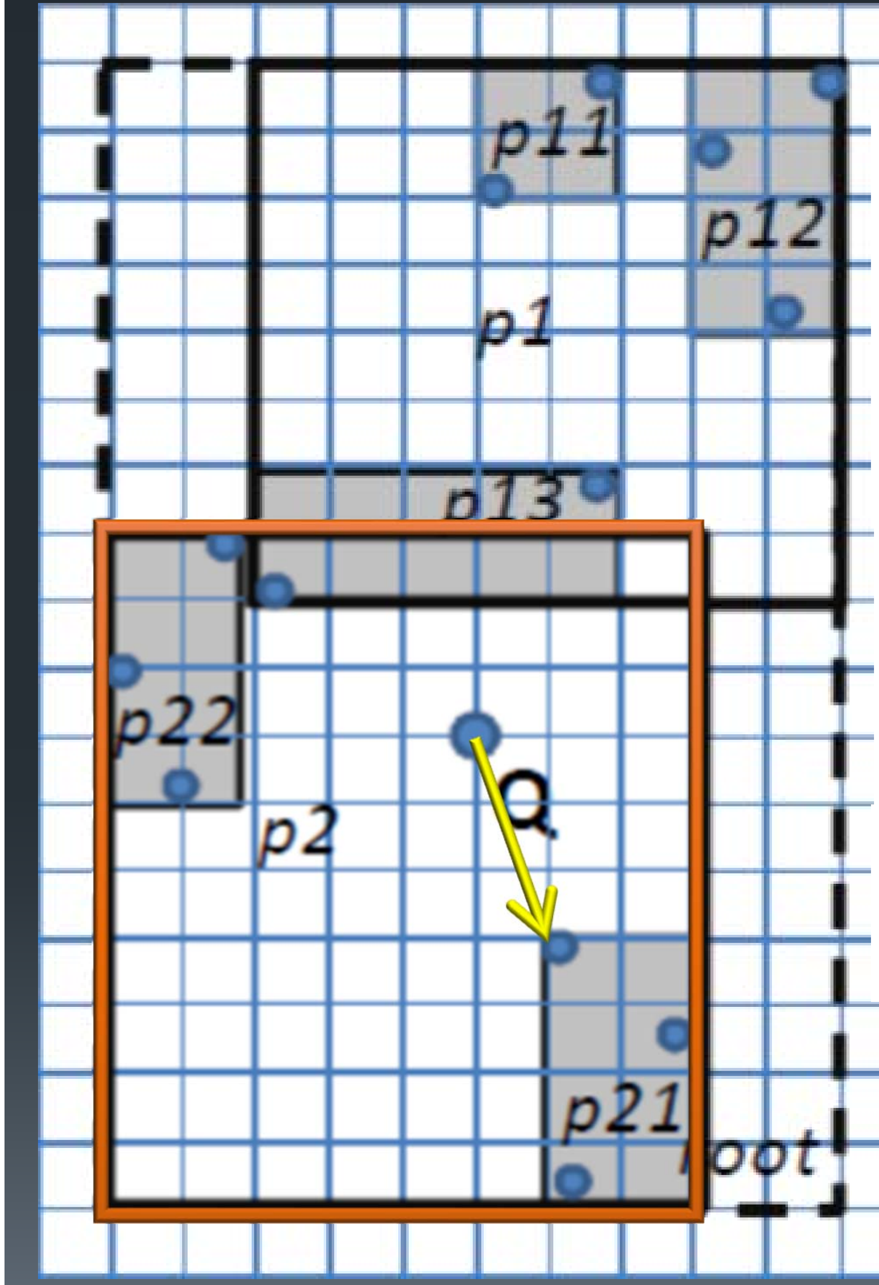
p22

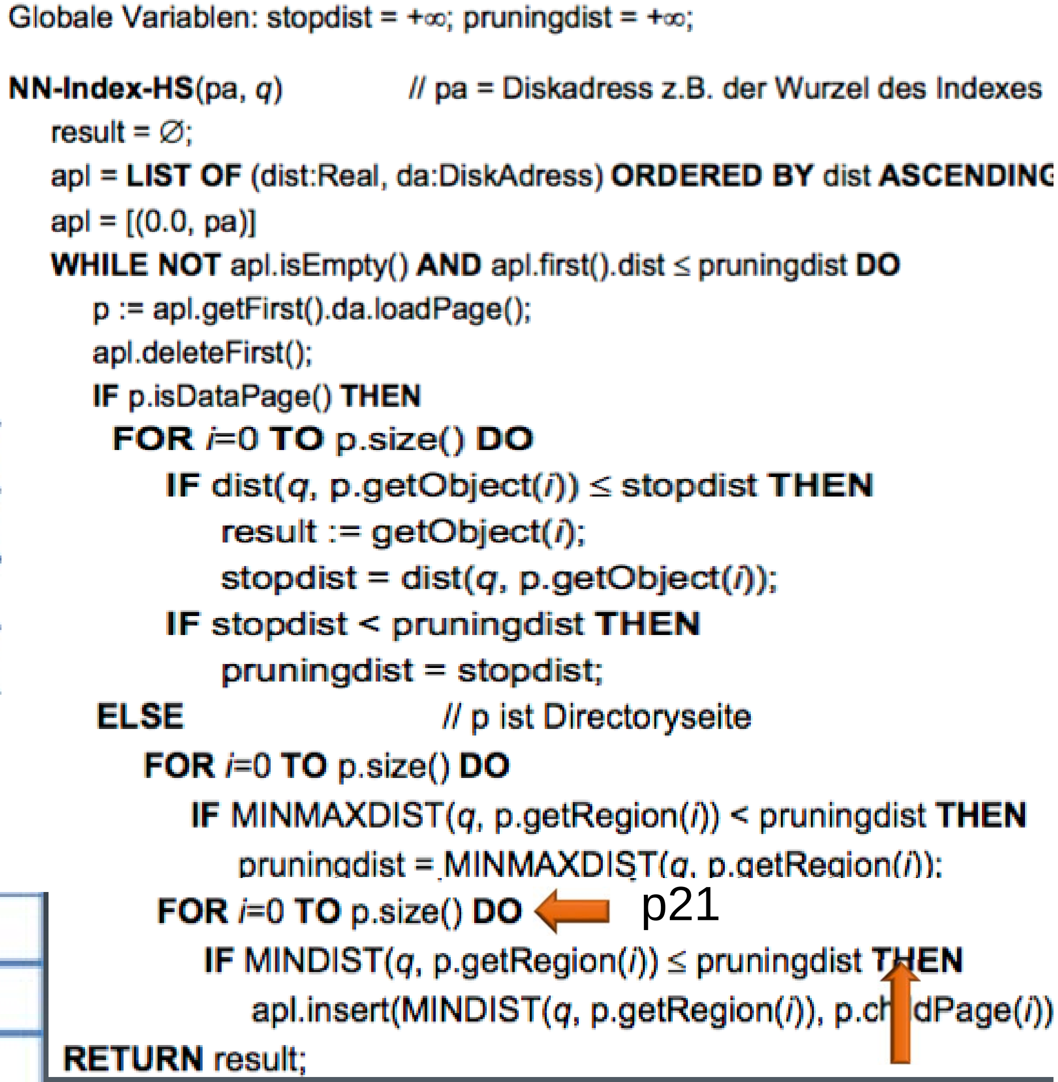
APL= [(2,p1)]

Besuchte Seiten: p2

stopdist = +Inf

pruningdist= 6





```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.ch dPage(i))
  RETURN result;
```

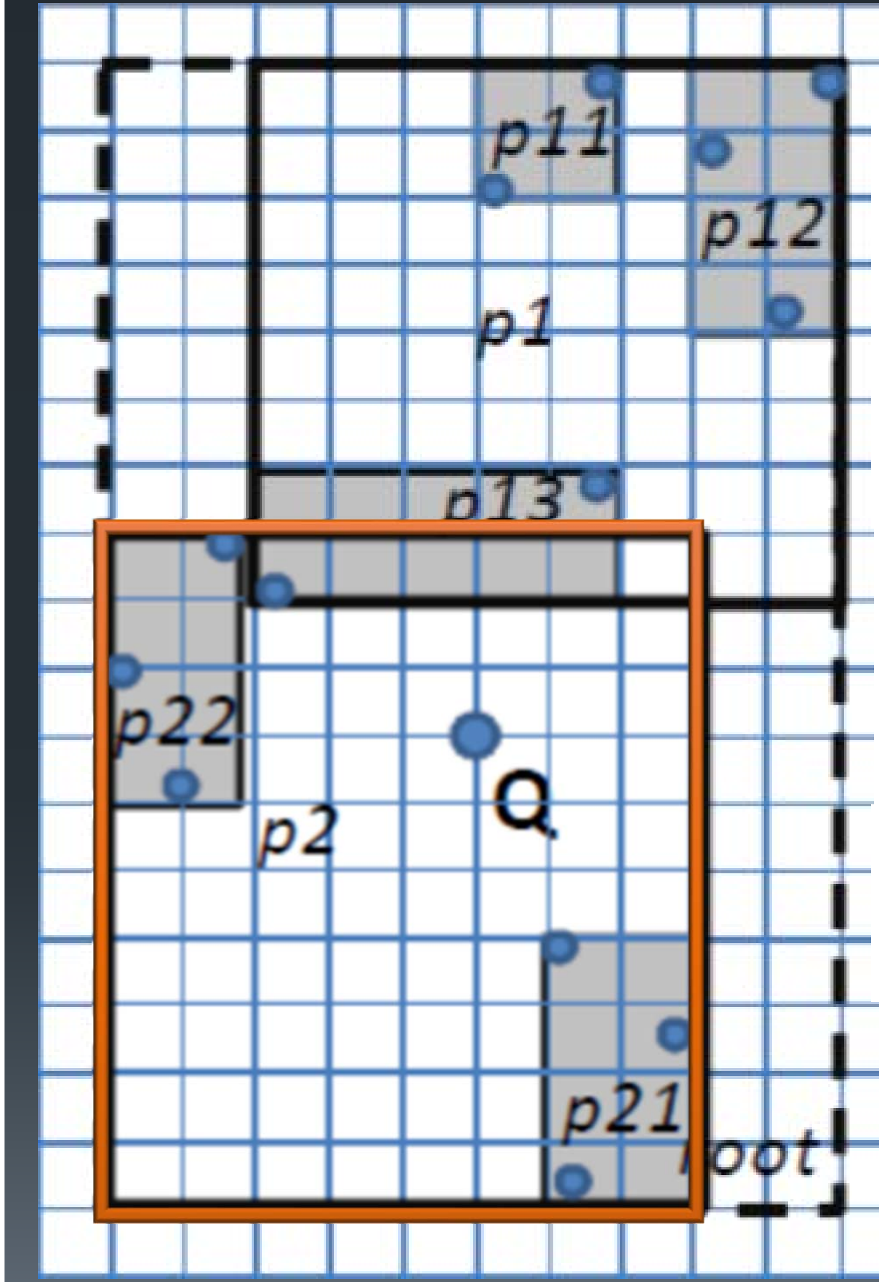
p21

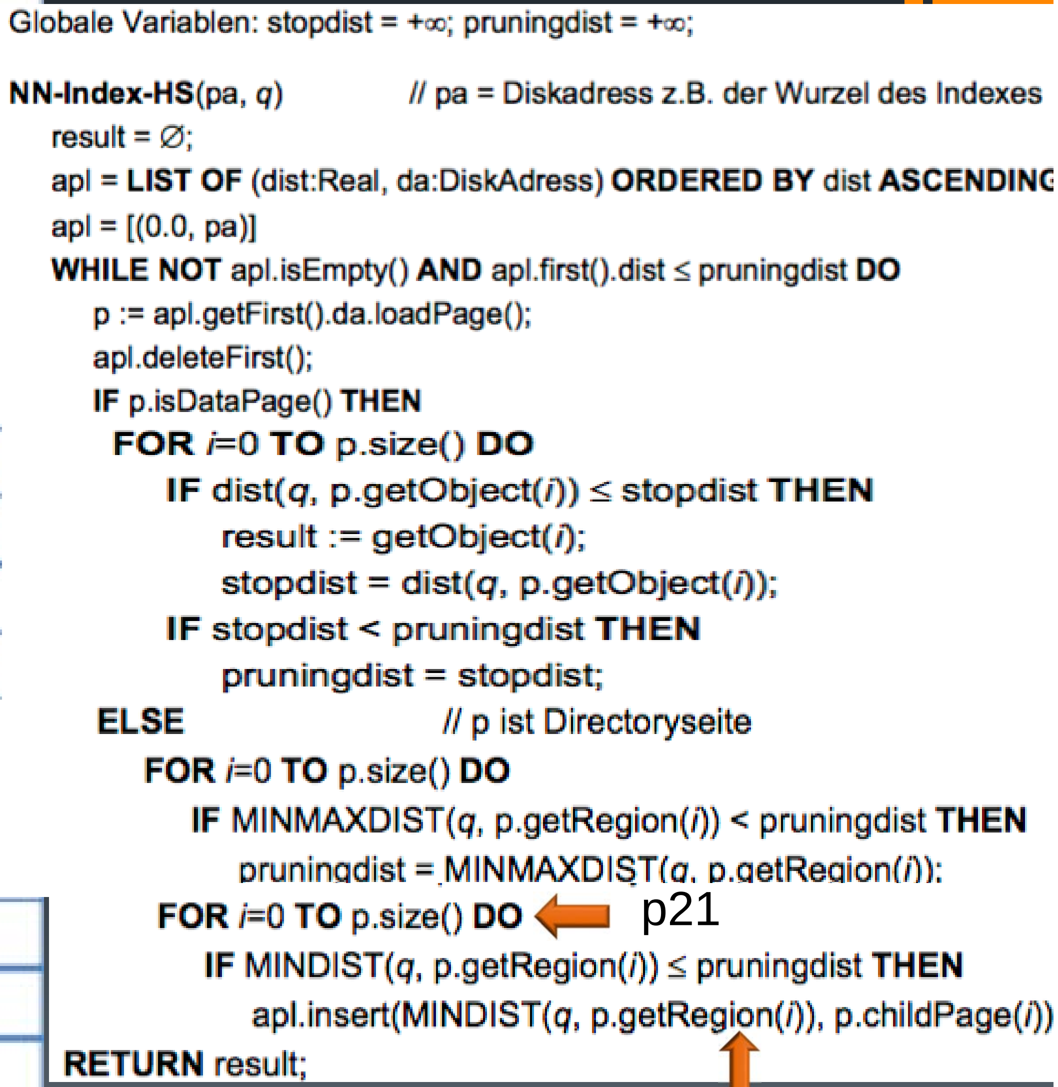
MINDIST(q,p21)= 4 <= 6

APL= [(2,p1)]

Besuchte Seiten: p2

stopdist = +Inf   pruningdist= 6



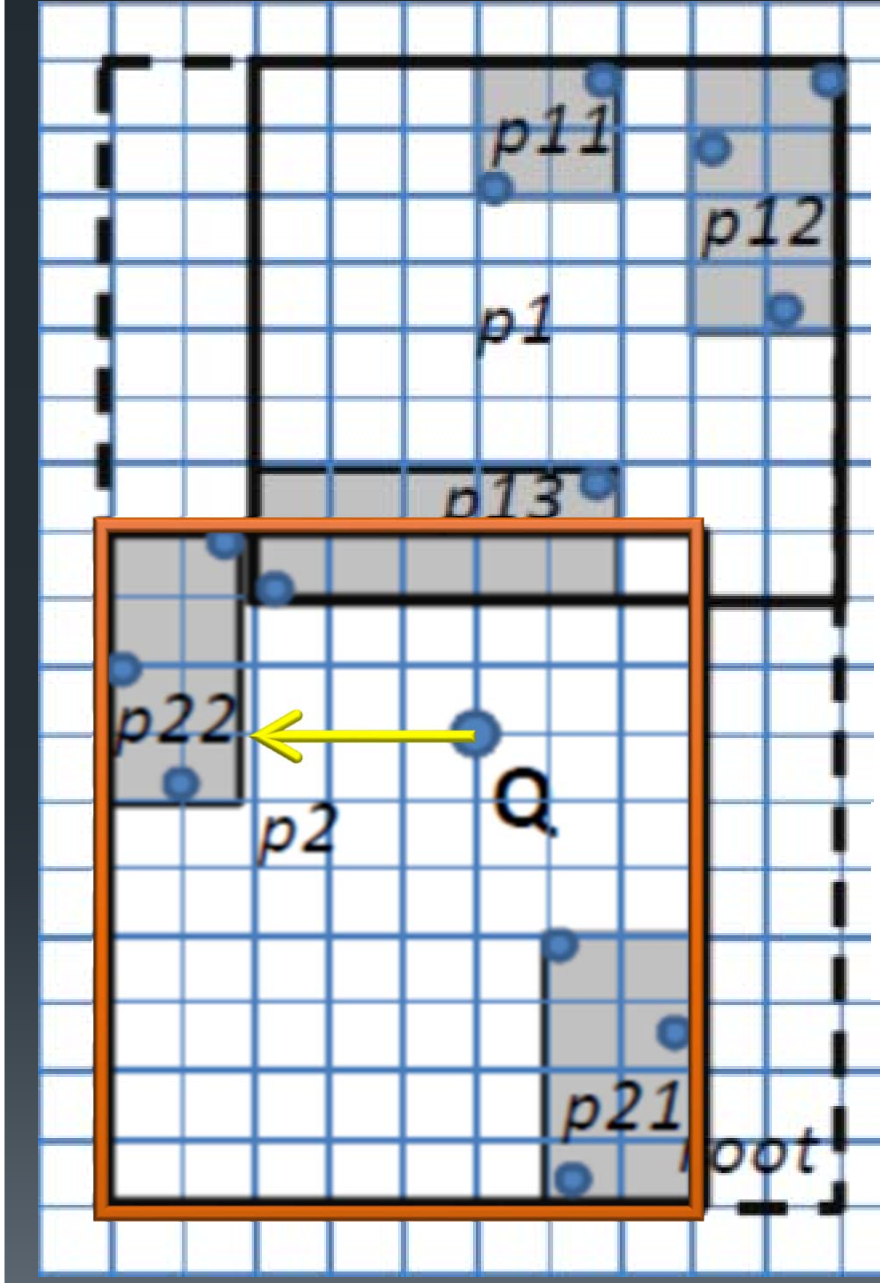


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

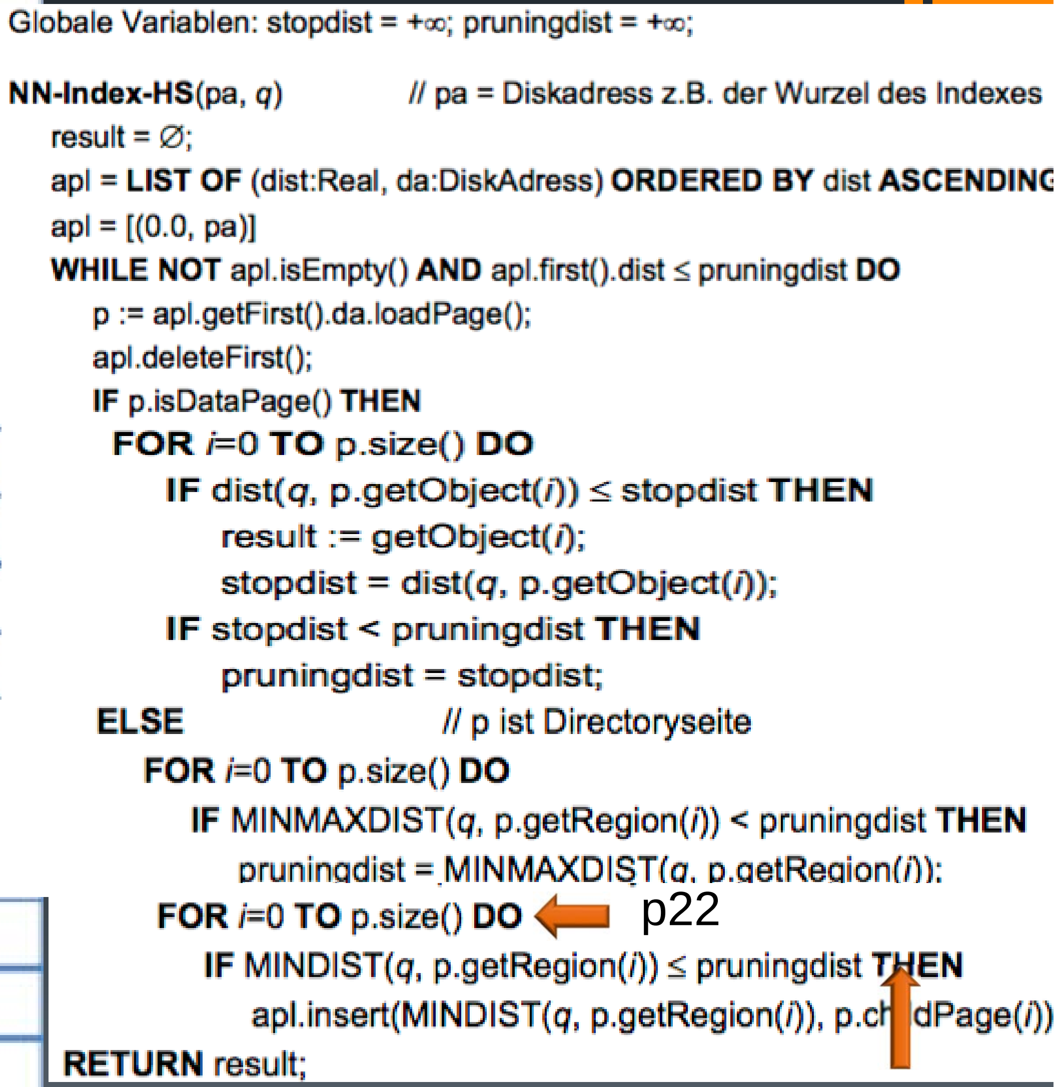
p21

APL= [(2,p1),(4,p21)]

Besuchte Seiten: p2



stopdist = +Inf pruningdist= 6

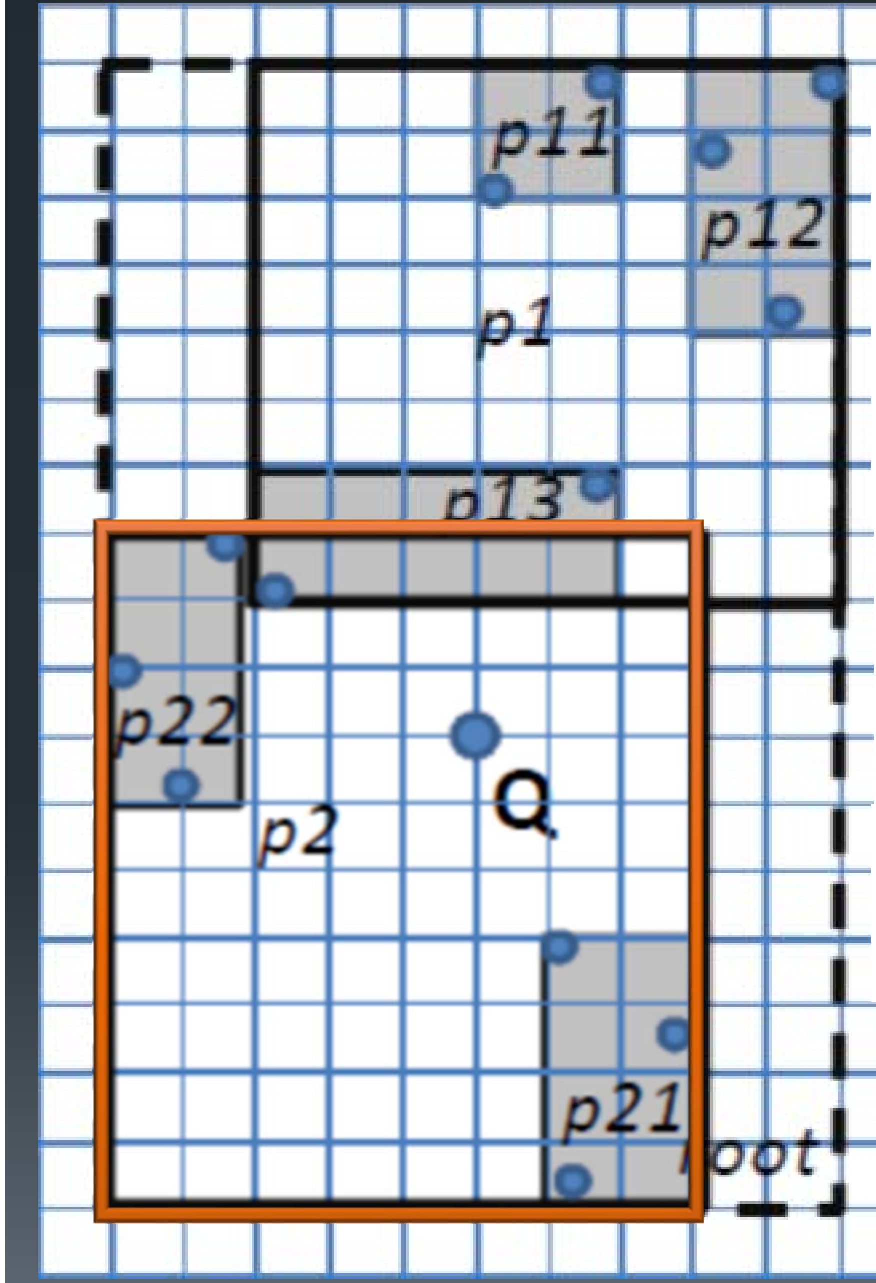


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO    p22
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

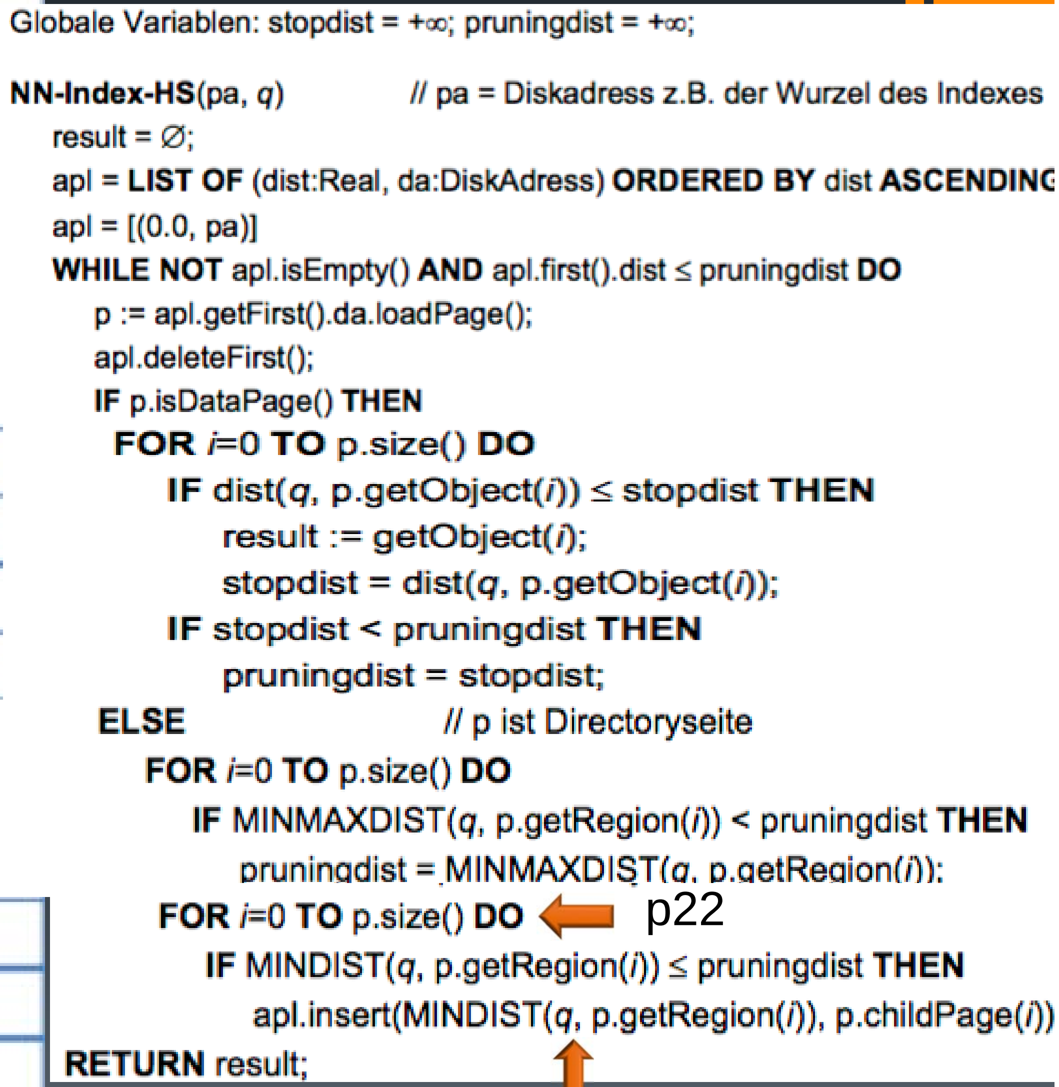
MINDIST(q,p22)= 3 <= 6

APL= [(2,p1),(4,p21)]

Besuchte Seiten: p2



stopdist = +Inf pruningdist= 6

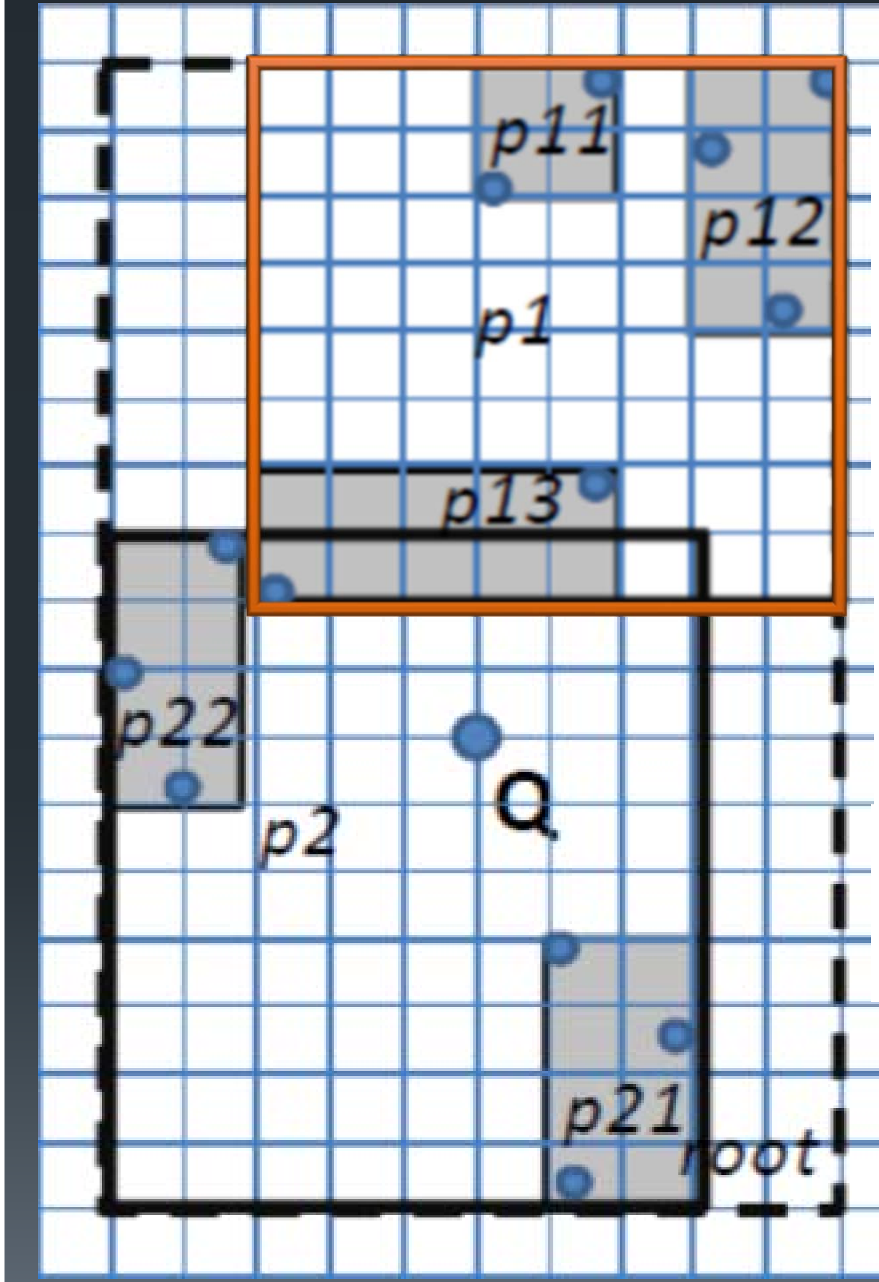


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

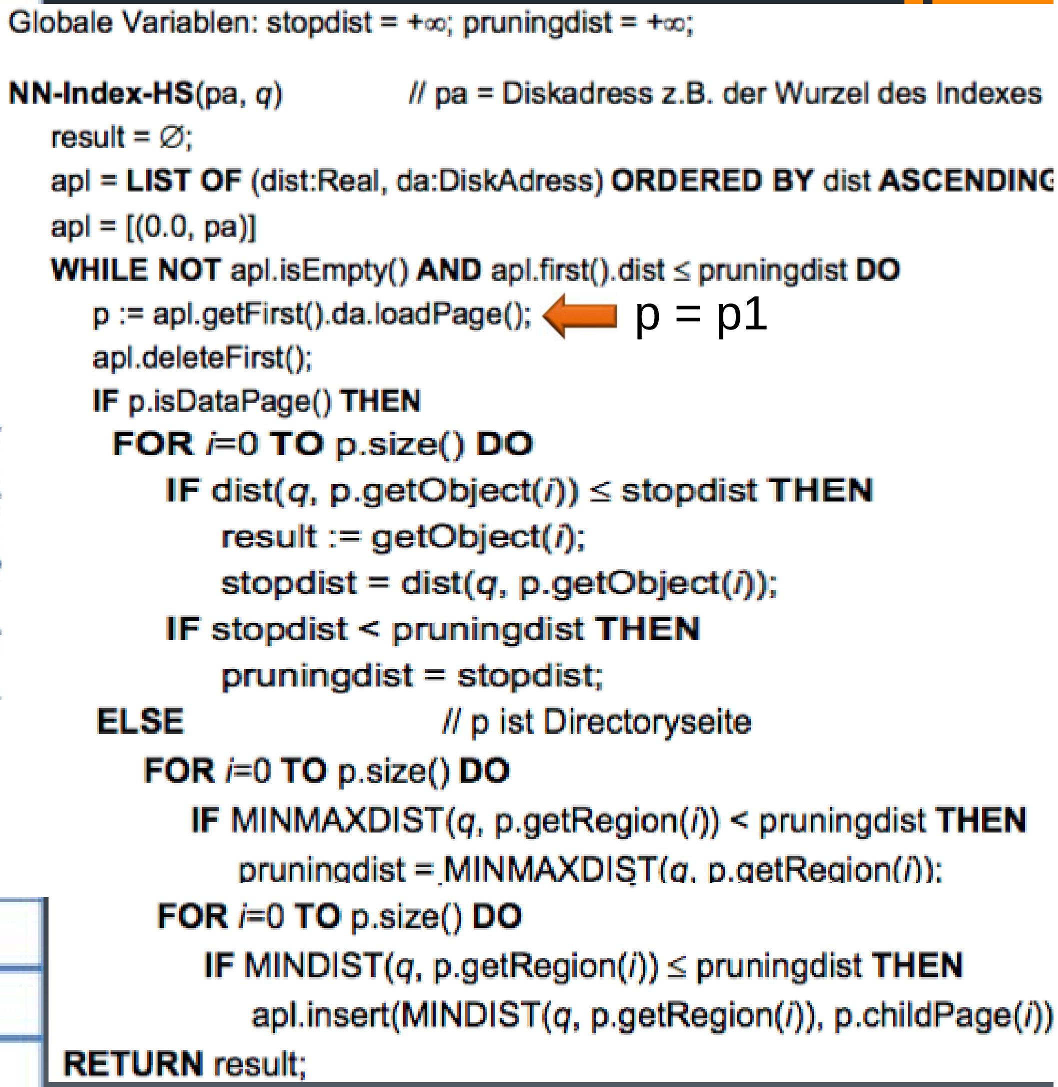
p22

APL= [(2,p1),(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf   pruningdist= 6

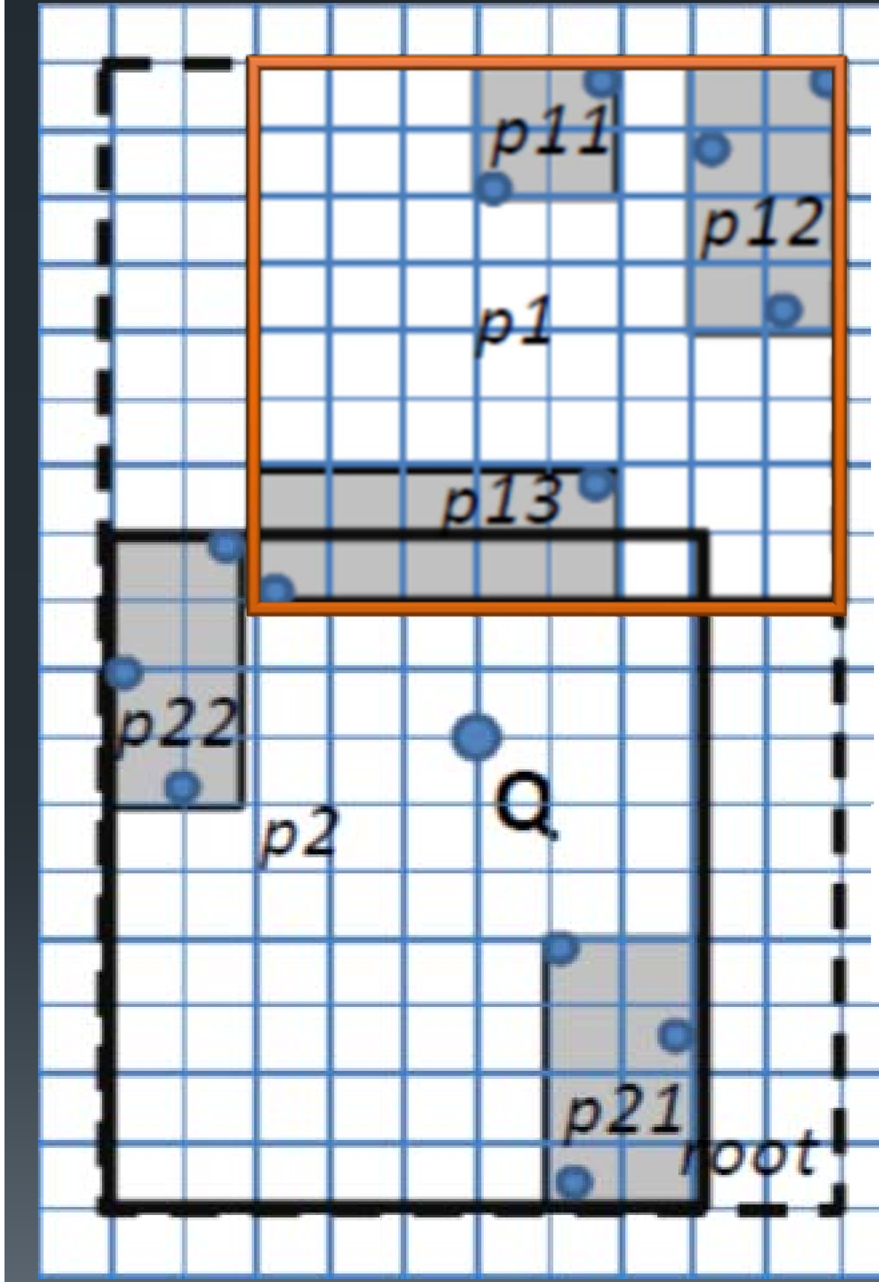


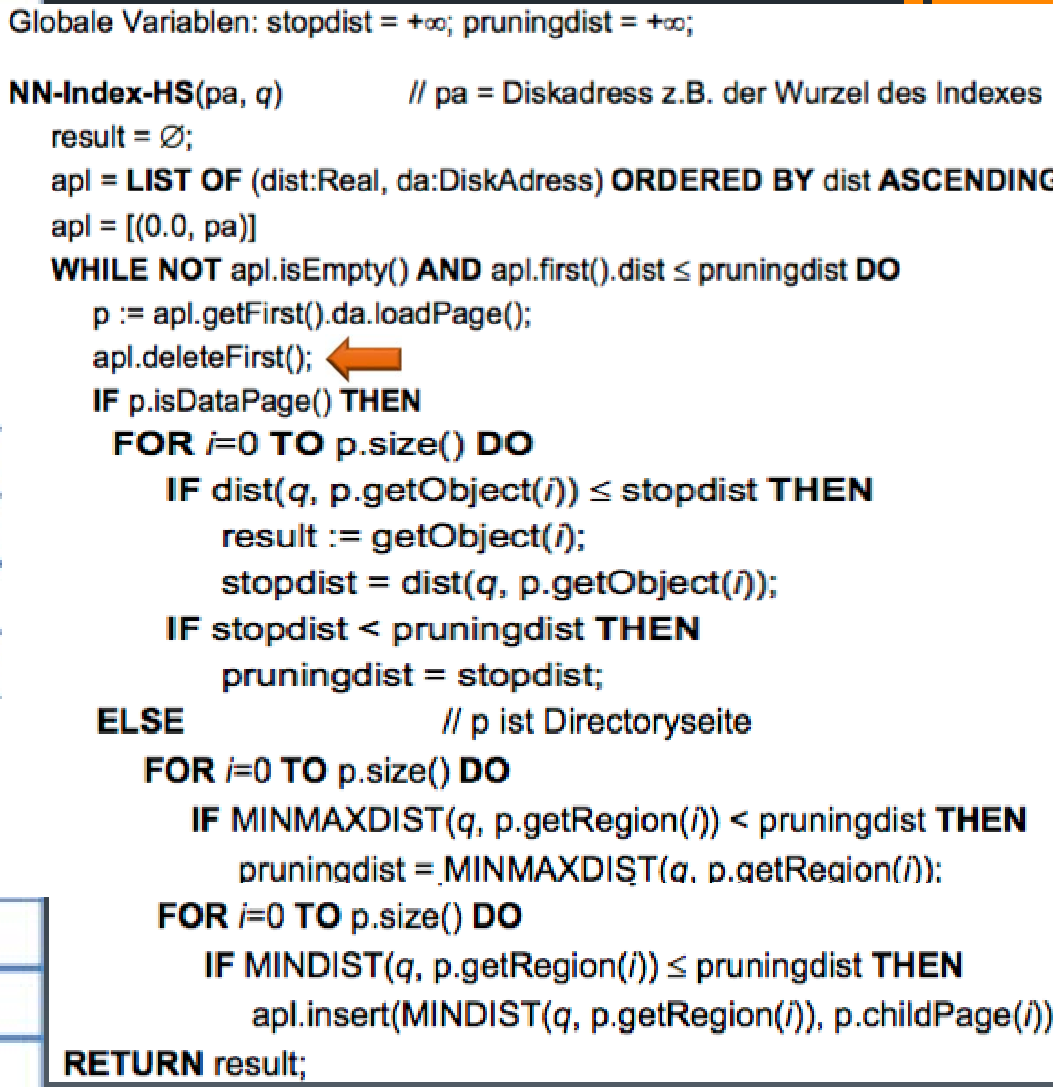
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();   ⇐ p = p1
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(2,p1),(3,p22),(4,p21)]

Besuchte Seiten: p2,p1

stopdist = +Inf   pruningdist= 6

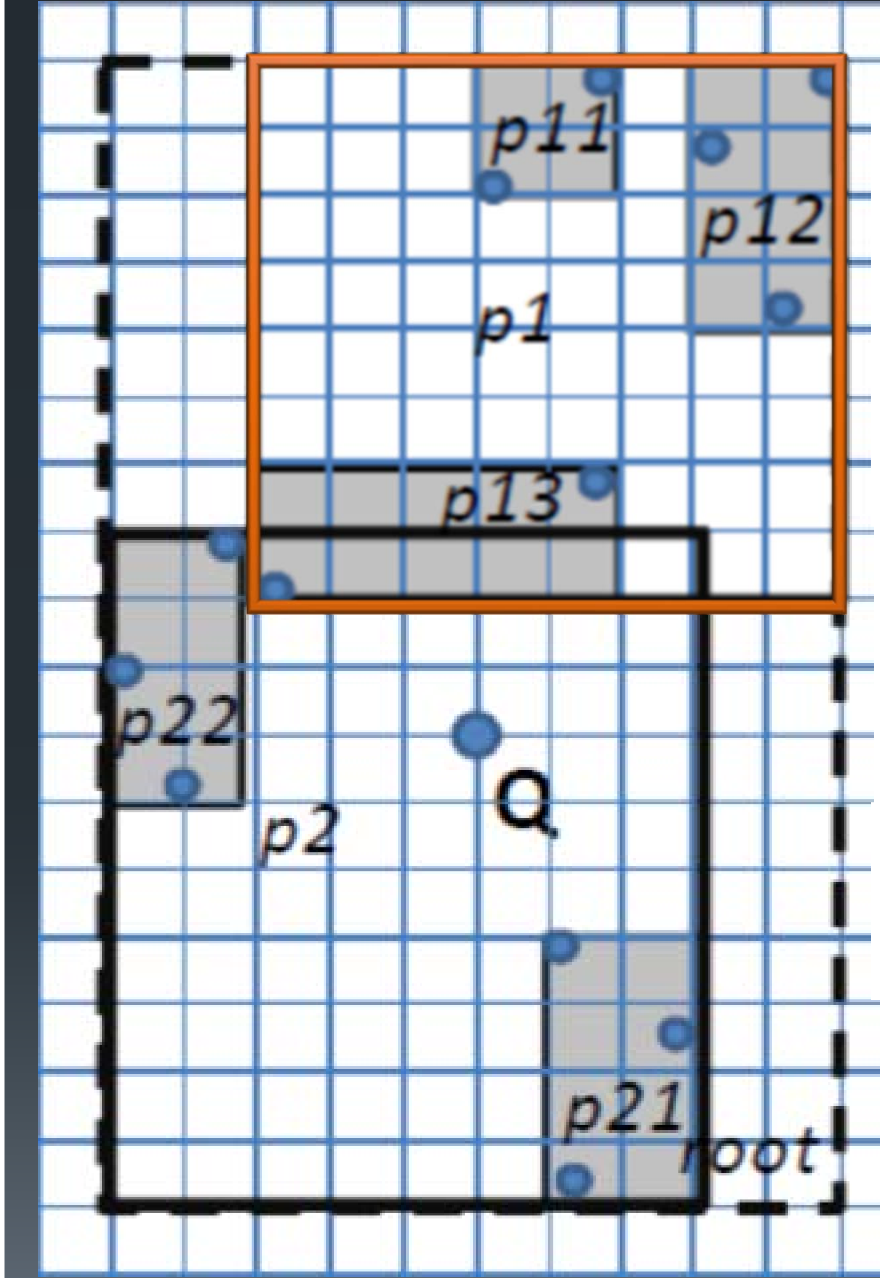




```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
           IF dist(q, p.getObject(i)) ≤ stopdist THEN
              result := getObject(i);
              stopdist = dist(q, p.getObject(i));
           IF stopdist < pruningdist THEN
              pruningdist = stopdist;
      ELSE                          // p ist Directoryseite
         FOR i=0 TO p.size() DO
            IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
               pruningdist = MINMAXDIST(q, p.getRegion(i));
         FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
               apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

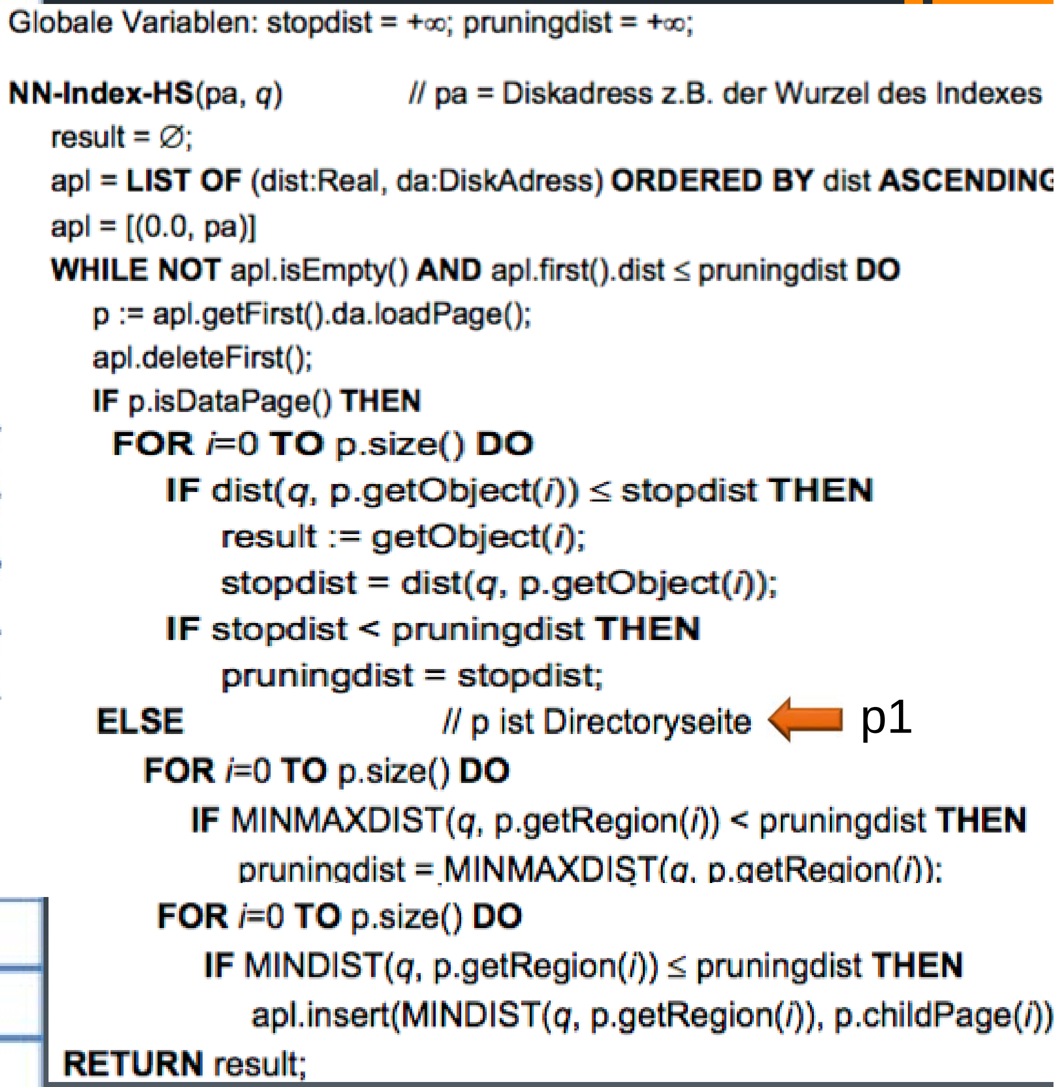
APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf

pruningdist= 6

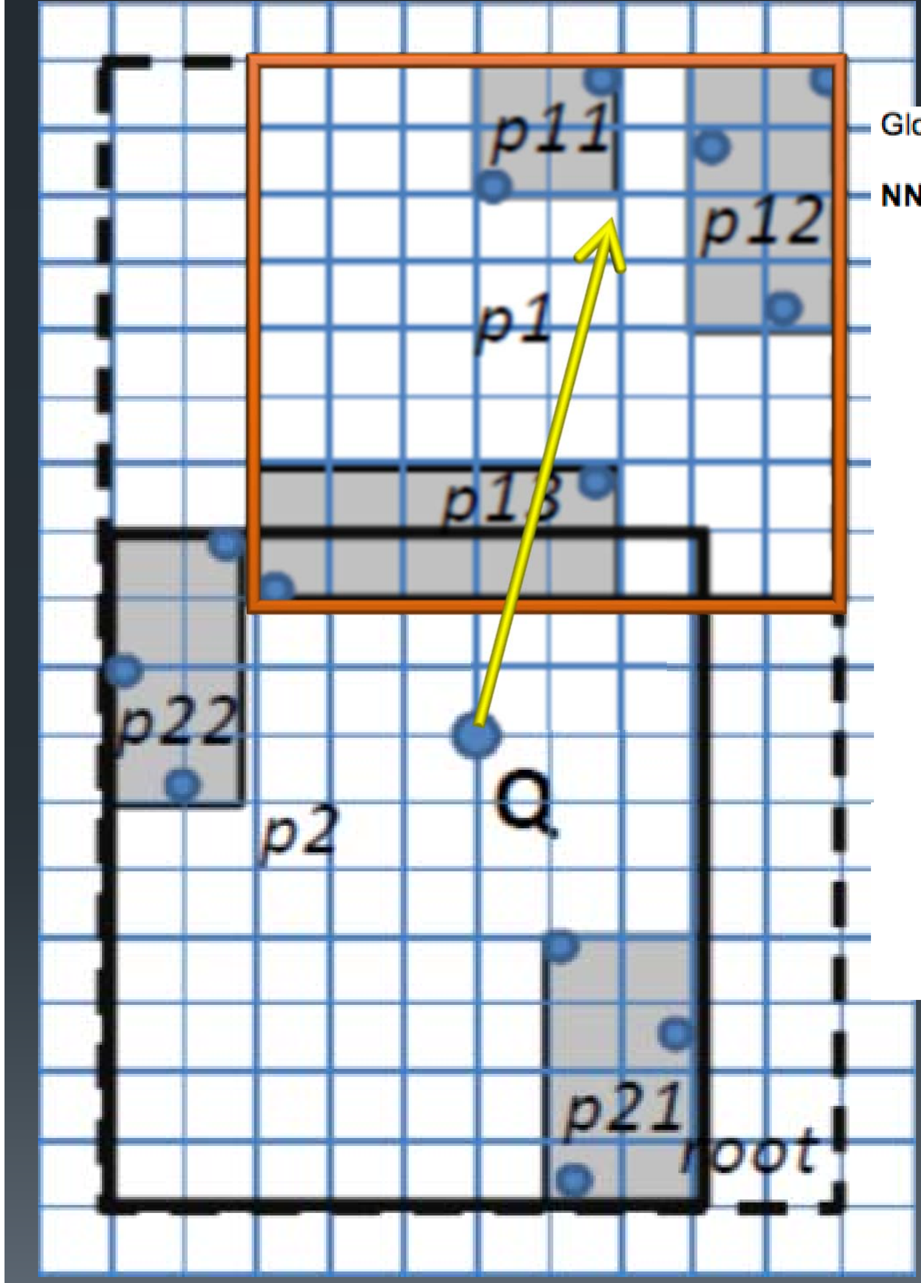


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

p1

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf pruningdist= 6

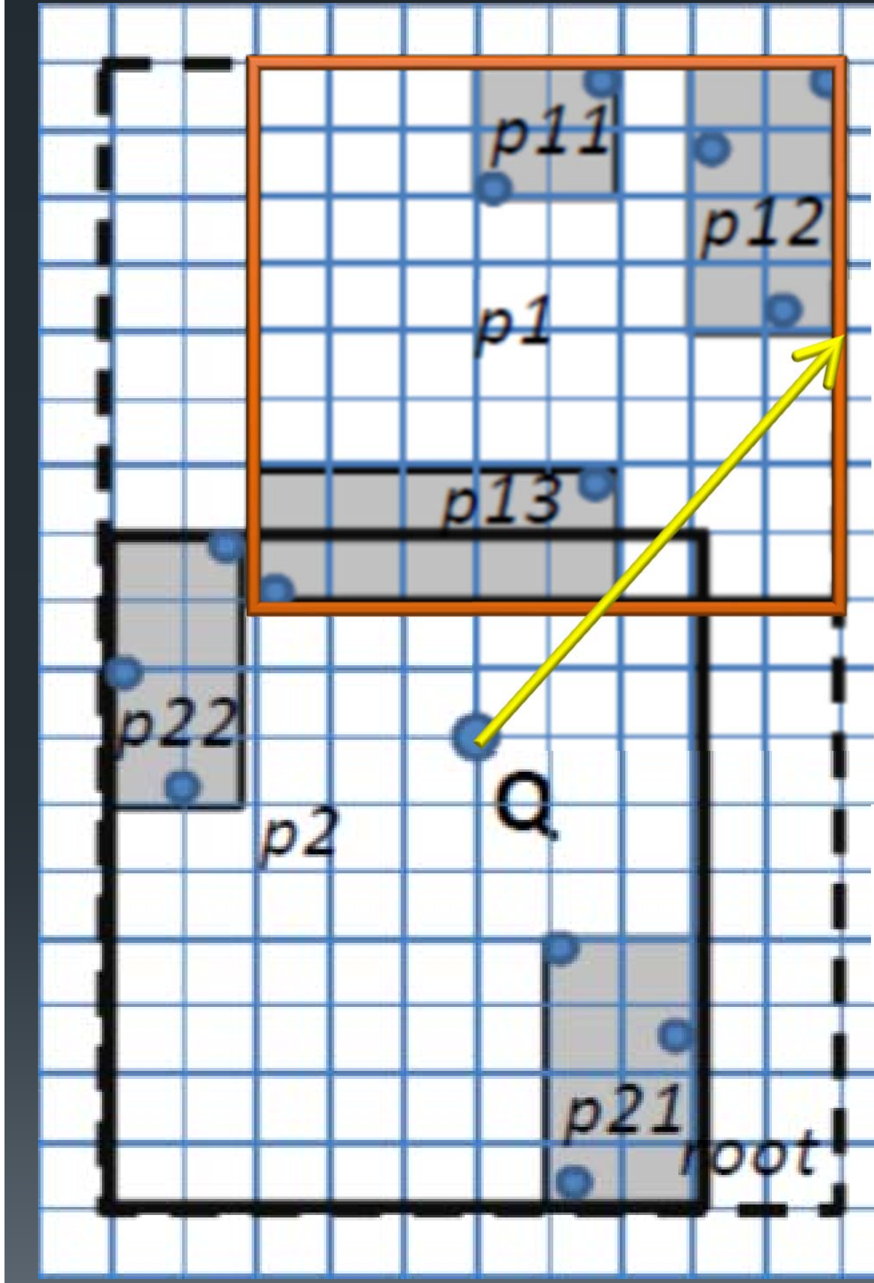
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDINC
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
       FOR i=0 TO p.size() DO
          IF dist(q, p.getObject(i)) ≤ stopdist THEN
             result := getObject(i);
             stopdist = dist(q, p.getObject(i));
          IF stopdist < pruningdist THEN
             pruningdist = stopdist;
      ELSE                        // p ist Directoryseite
        FOR i=0 TO p.size() DO
           IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
              pruningdist = MINMAXDIST(q, p.getRegion(i));
        FOR i=0 TO p.size() DO
           IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
              apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

10 > 6

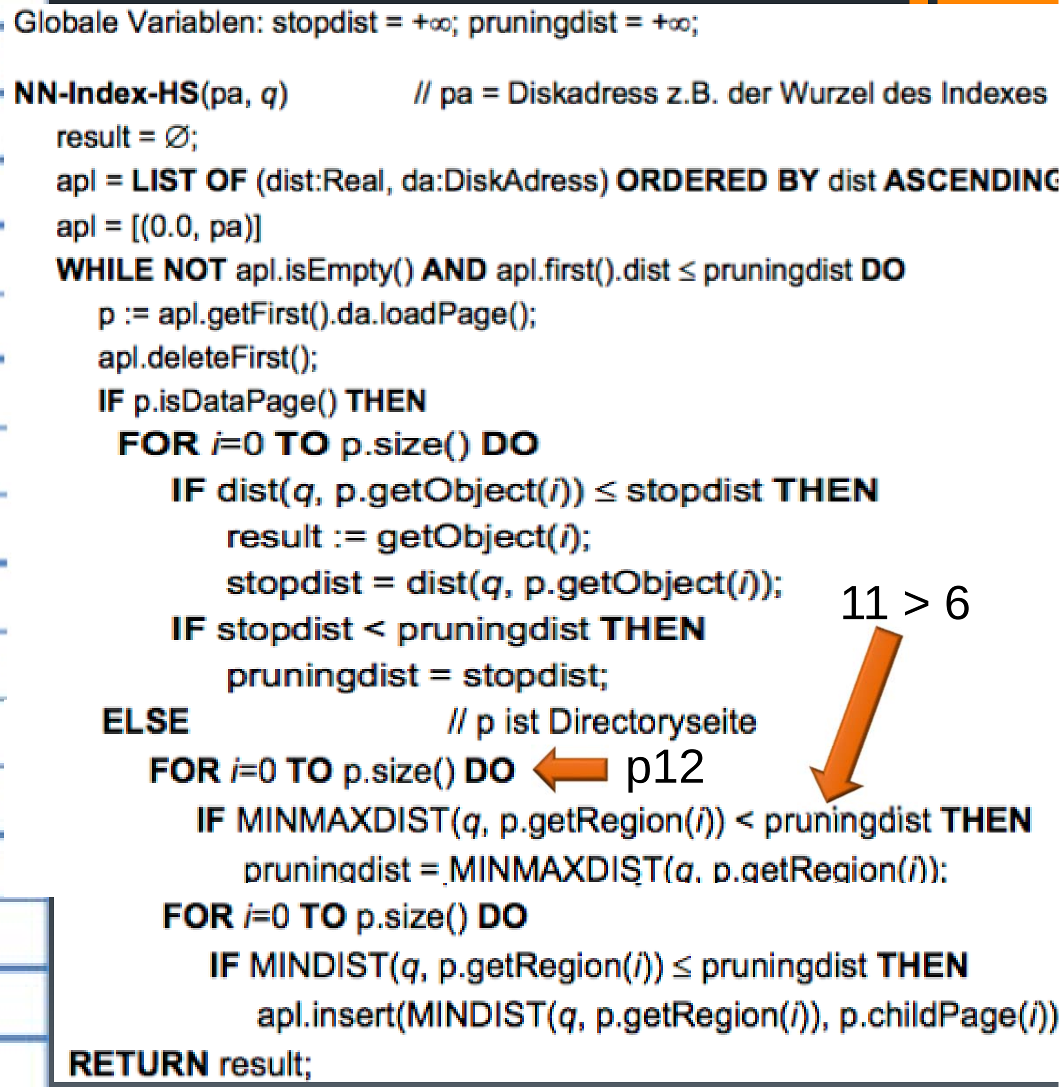
p11

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf pruningdist= 6



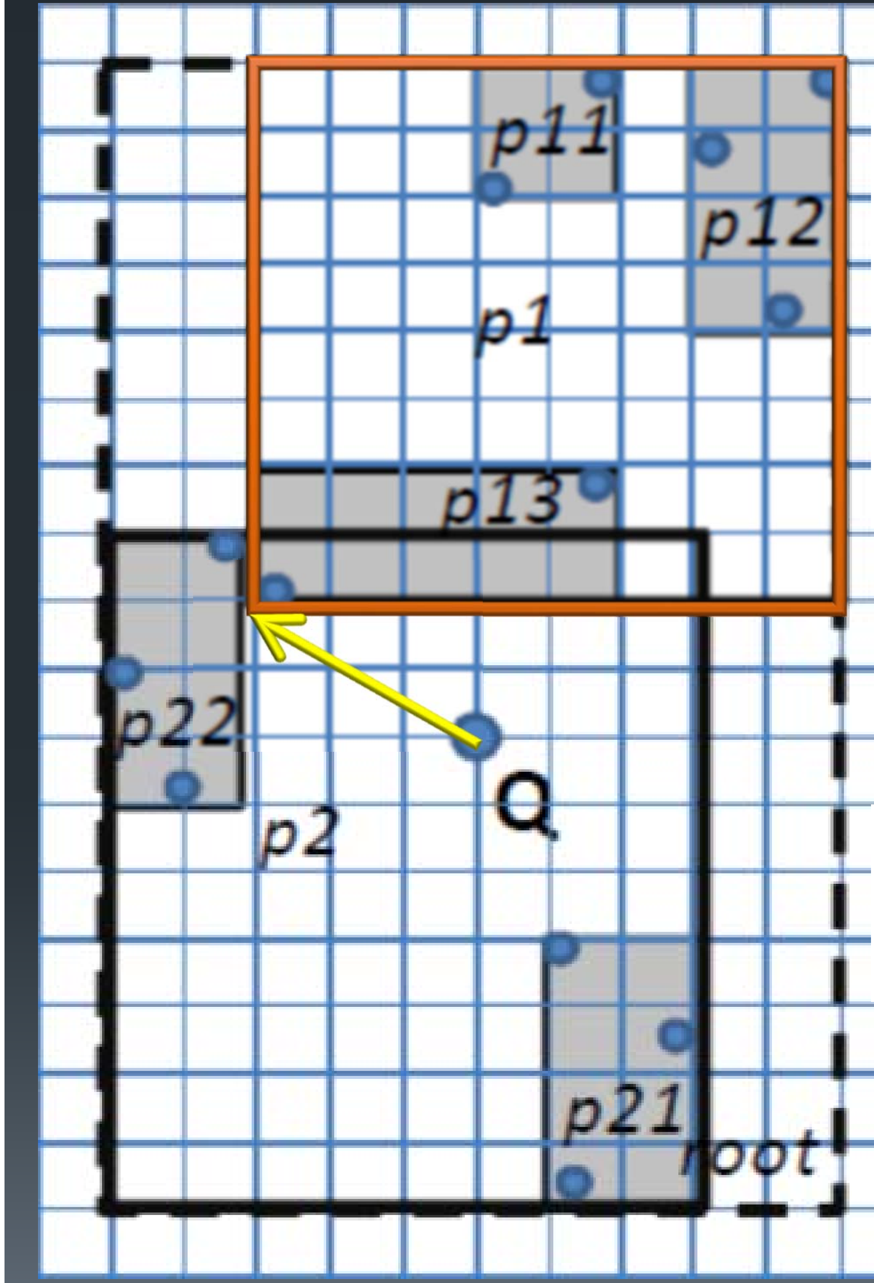
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
       FOR i=0 TO p.size() DO
          IF dist(q, p.getObject(i)) ≤ stopdist THEN
             result := getObject(i);
             stopdist = dist(q, p.getObject(i));
          IF stopdist < pruningdist THEN
             pruningdist = stopdist;
      ELSE                        // p ist Directoryseite
         FOR i=0 TO p.size() DO
            IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
               pruningdist = MINMAXDIST(q, p.getRegion(i));
         FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
               apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

11 > 6

p12

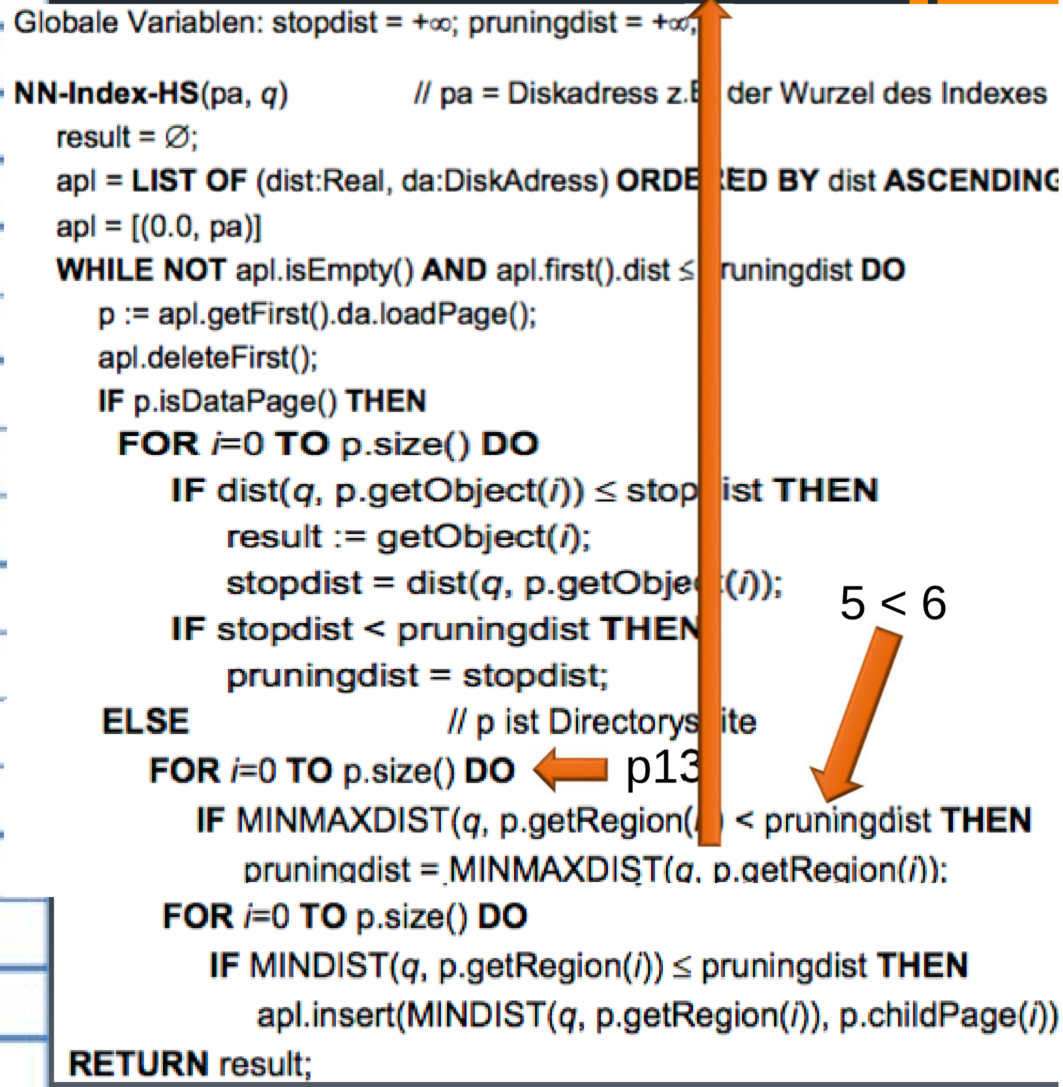
APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf

pruningdist= 5



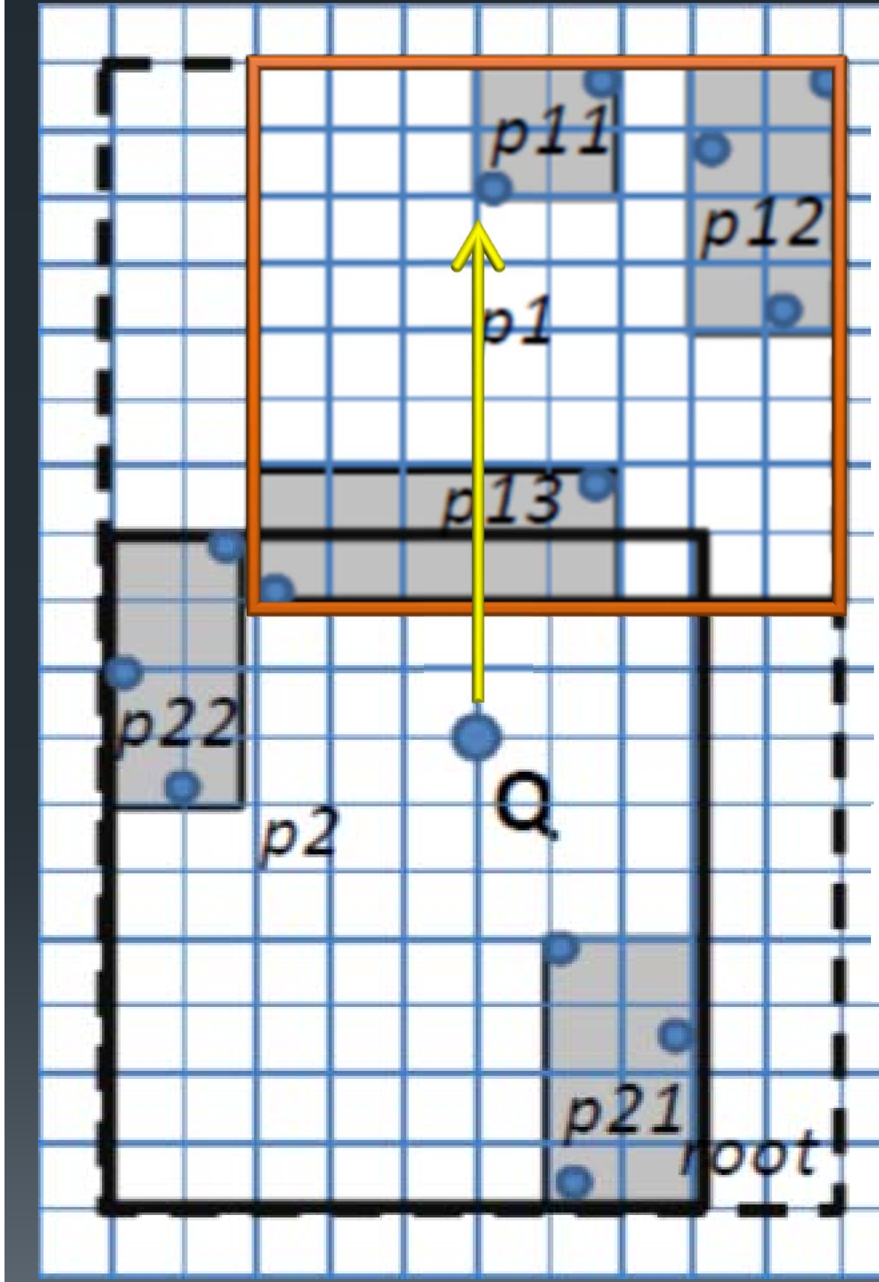
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

p13

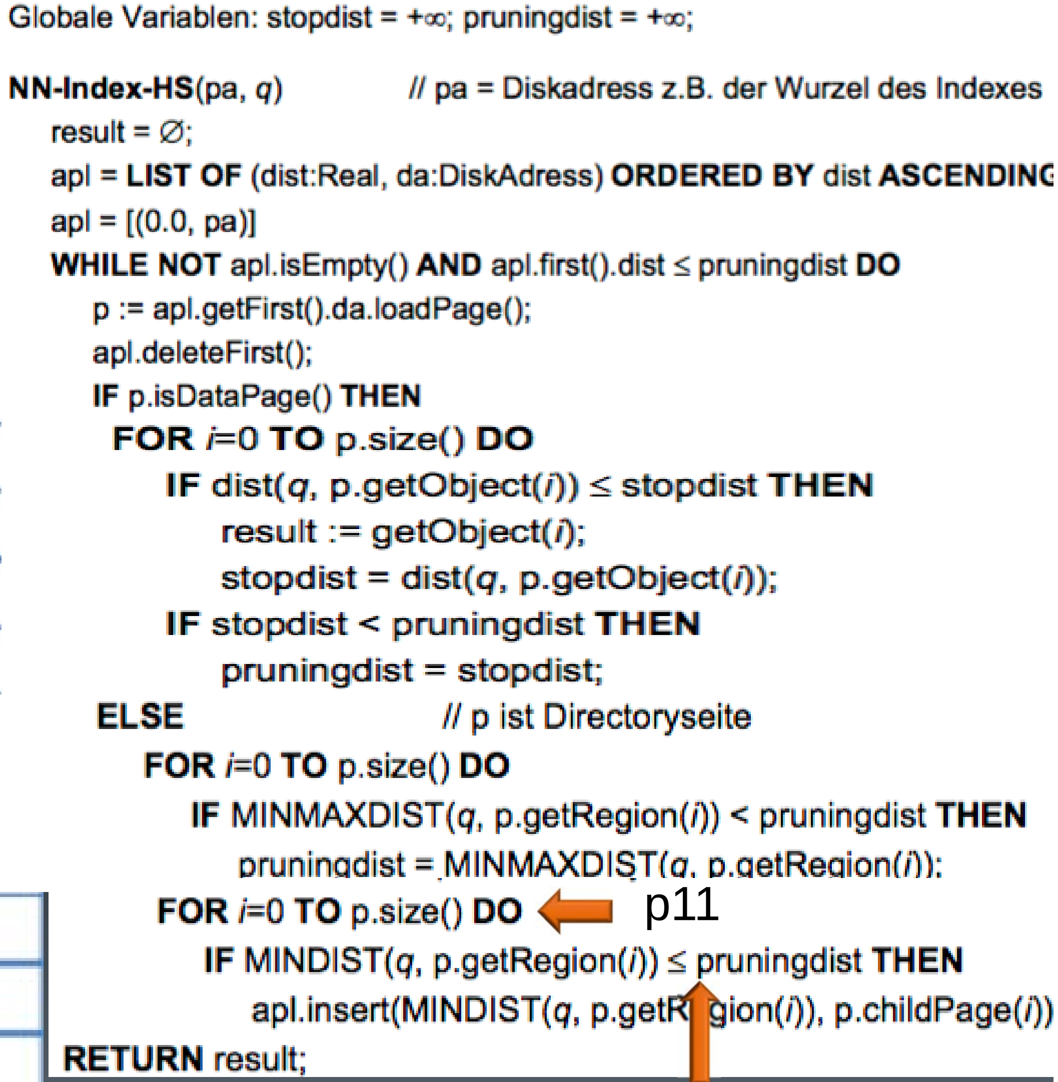
5 < 6

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf   pruningdist= 5



```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

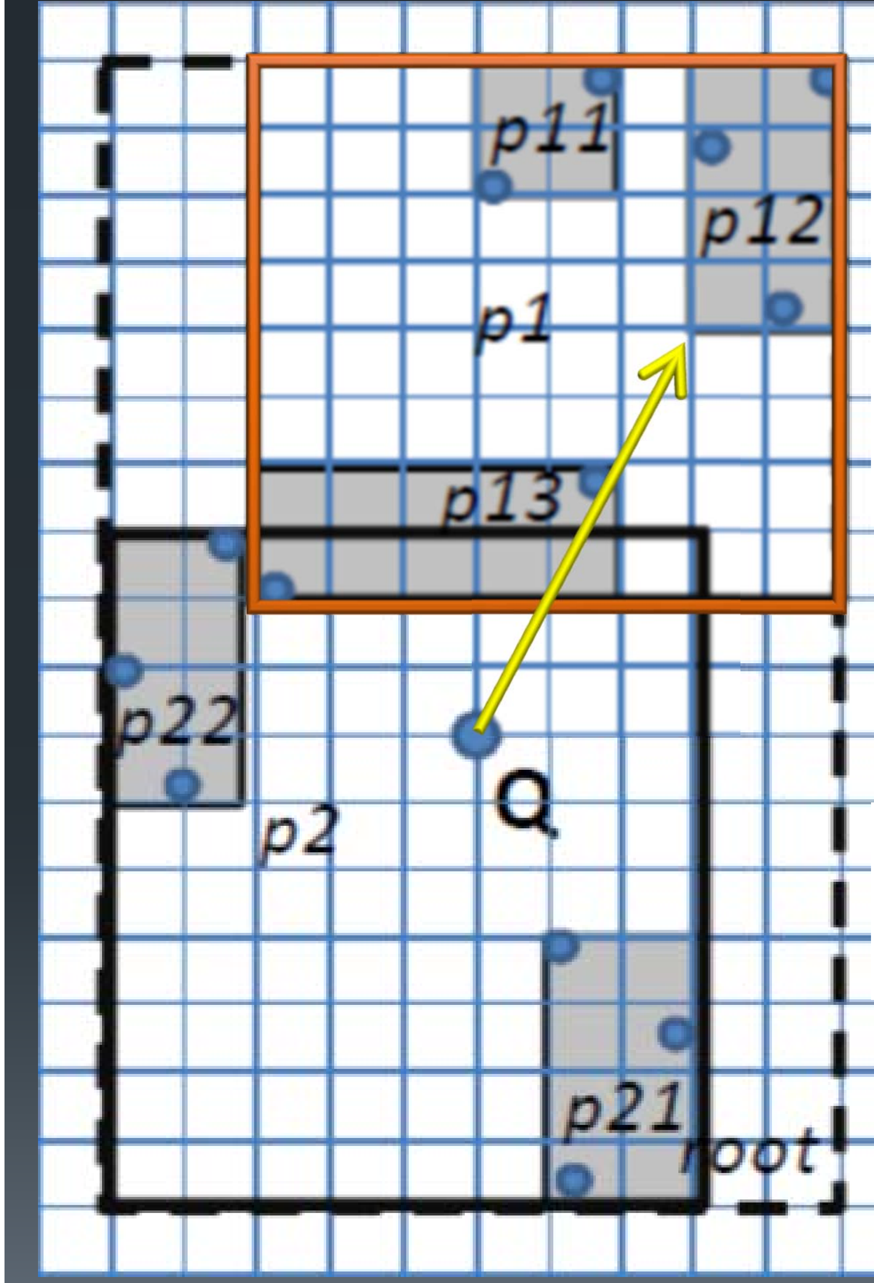
p11

MINDIST(q,p11)= 8 > 5

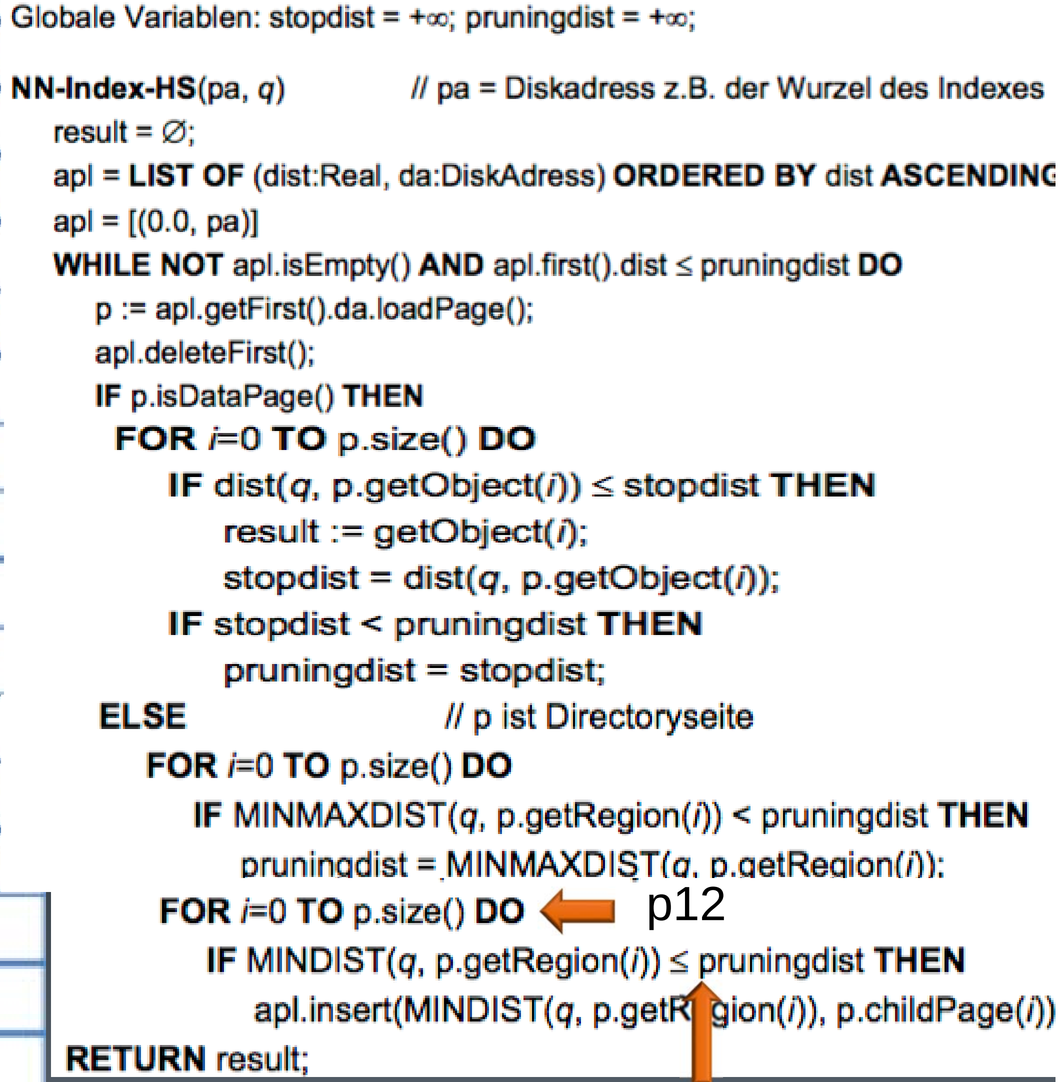
PRUNE p11

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1



stopdist = +Inf   pruningdist= 5



```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
           result := getObject(i);
           stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
           pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
         IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
           pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
         IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
           apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

p12

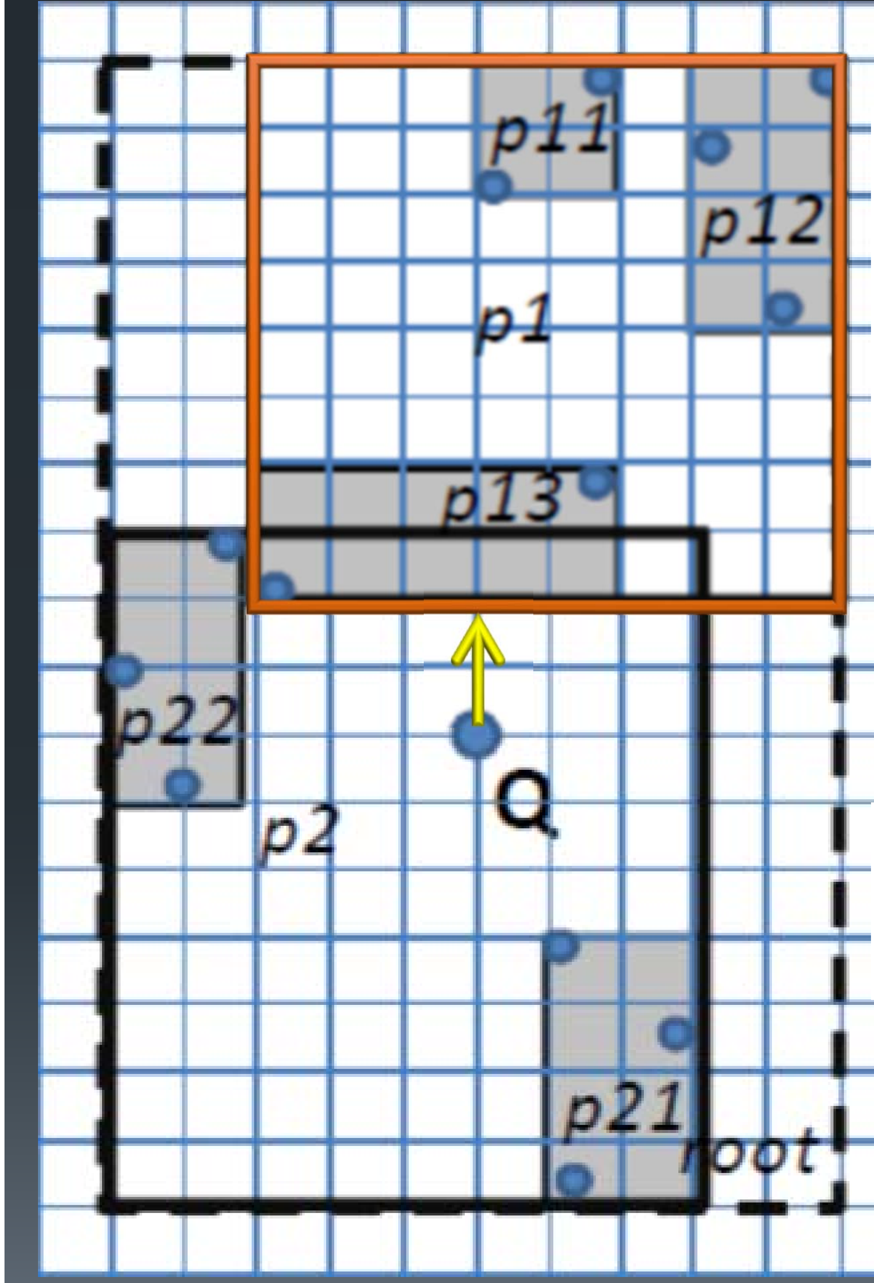
MINDIST(q,p12)= 9 > 5

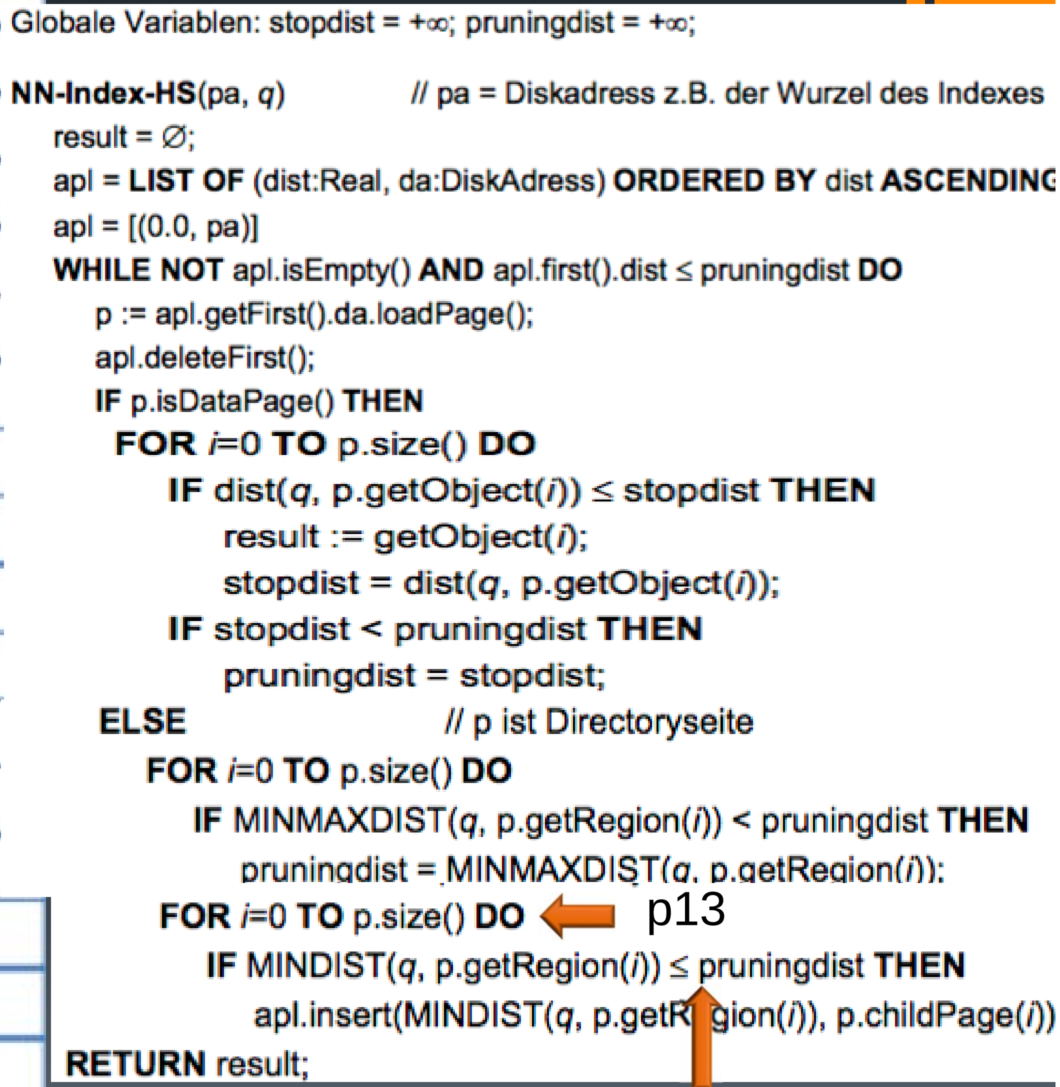
APL= [(3,p22),(4,p21)]

PRUNE p12

Besuchte Seiten: p2,p1

stopdist = +Inf pruningdist= 5





```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                      // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

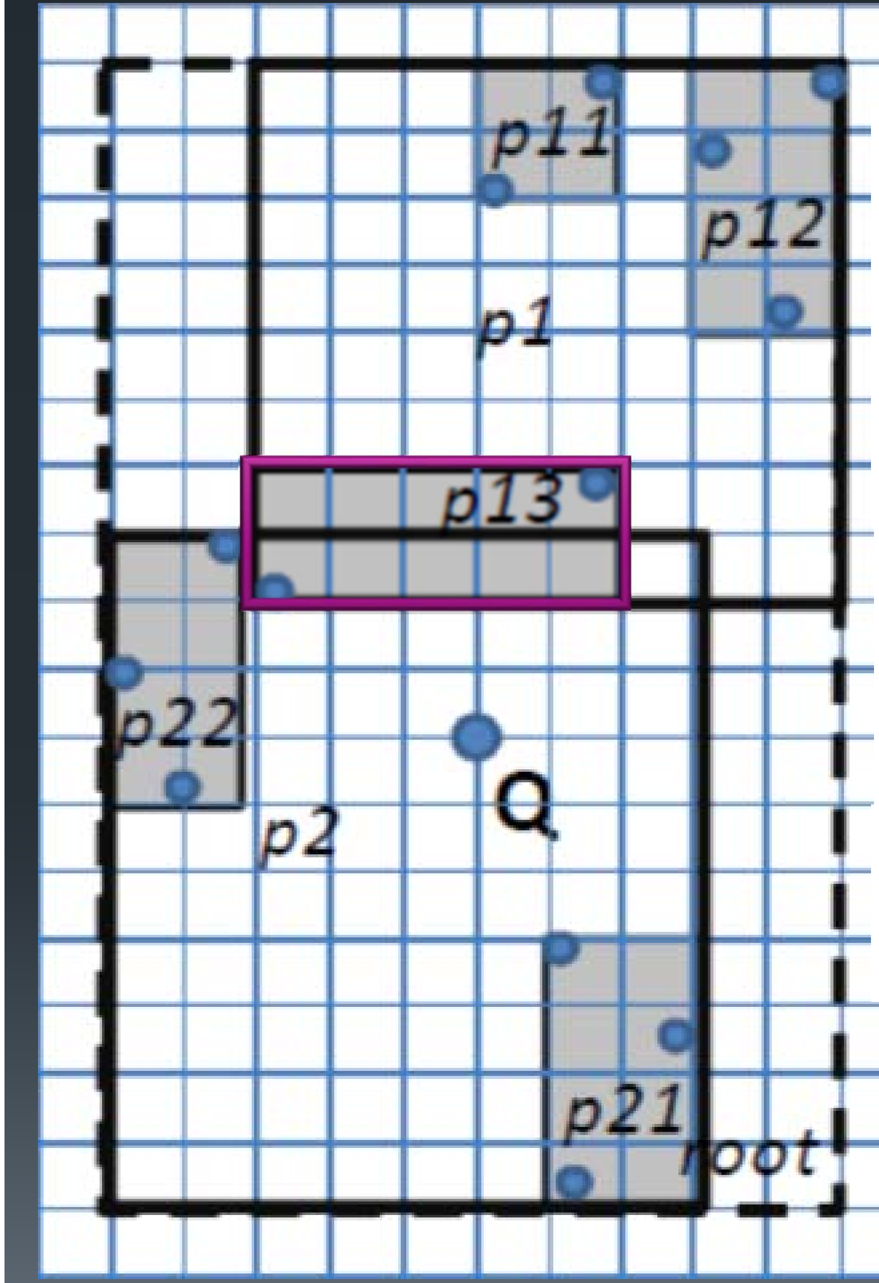
p13

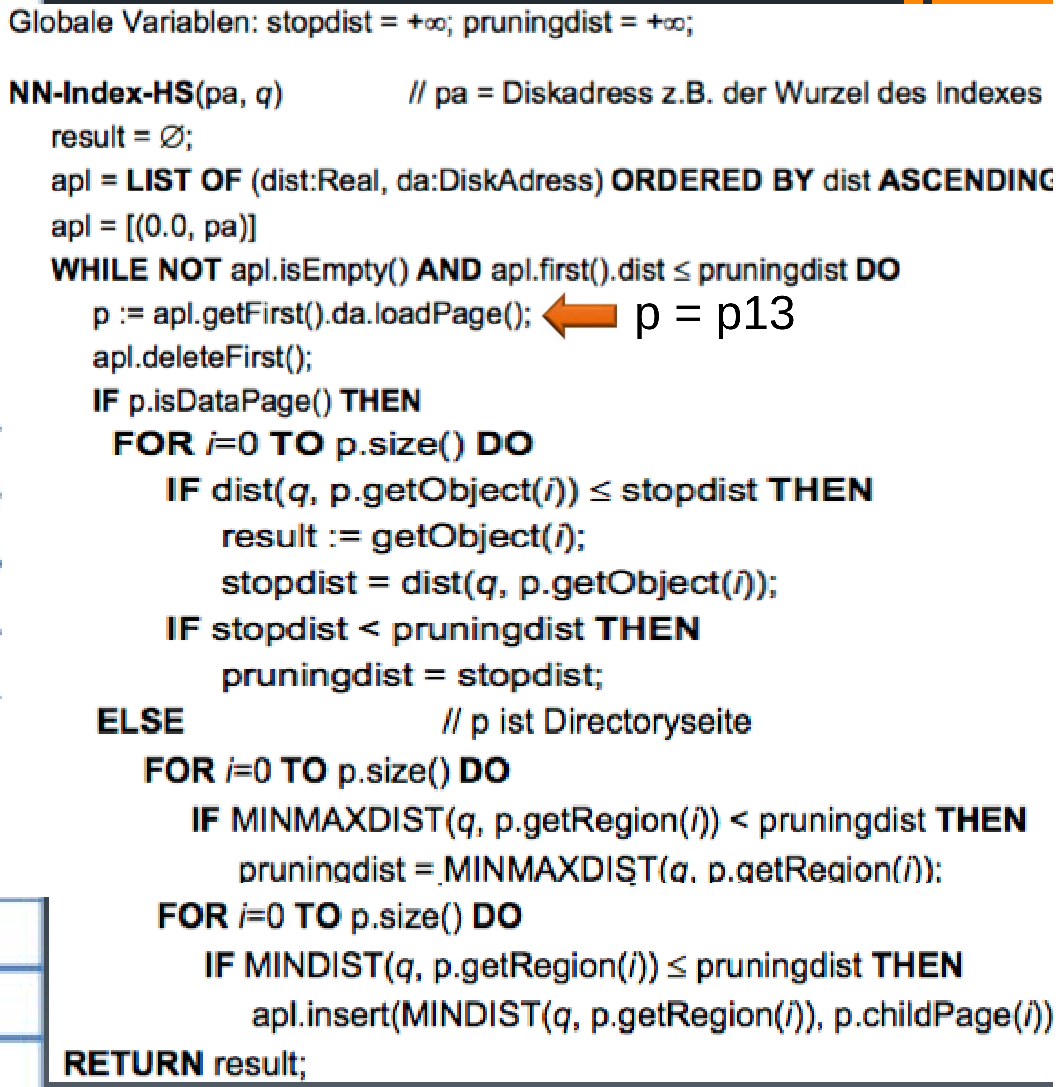
MINDIST(q,p13)= 2 < 5

APL= [(2,p13),(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13

stopdist = +Inf   pruningdist= 5





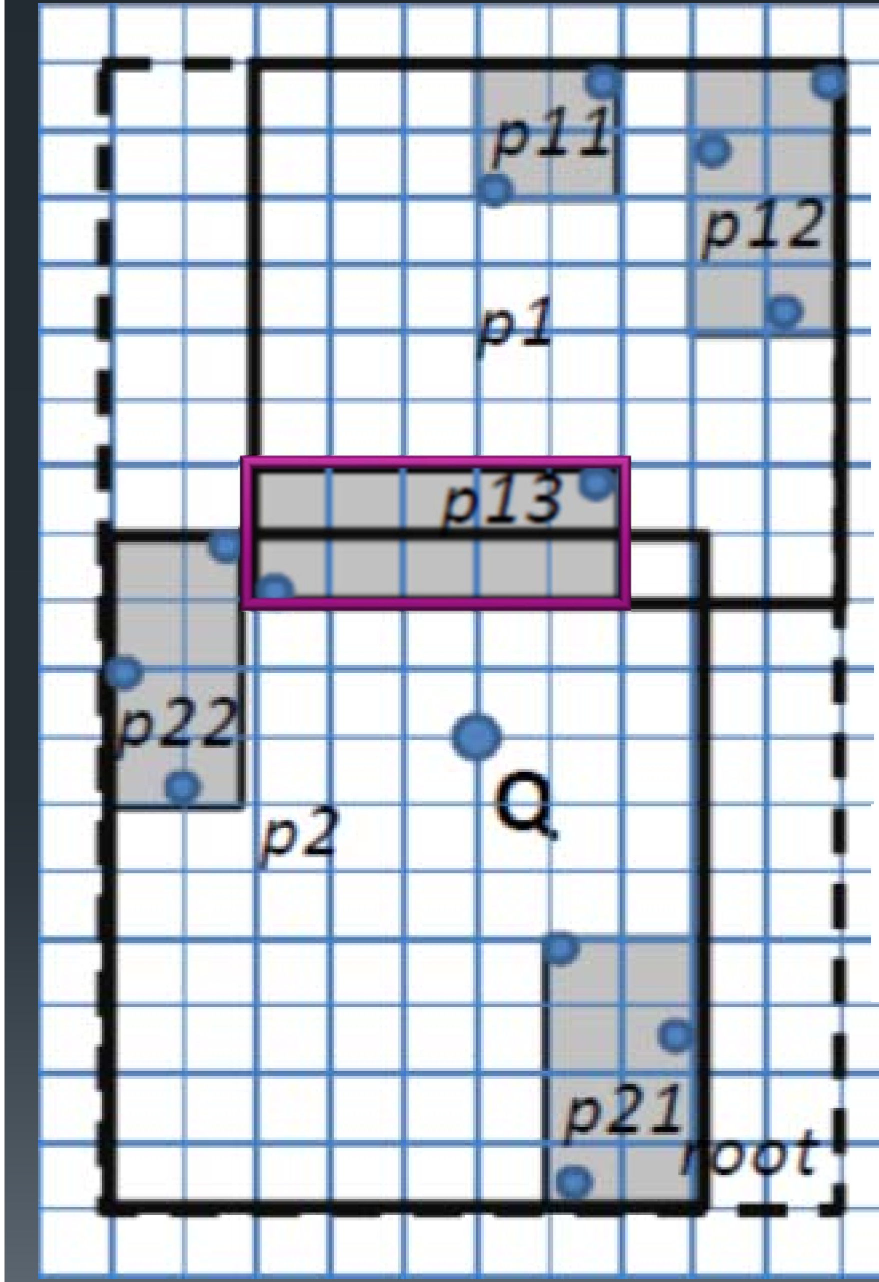
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

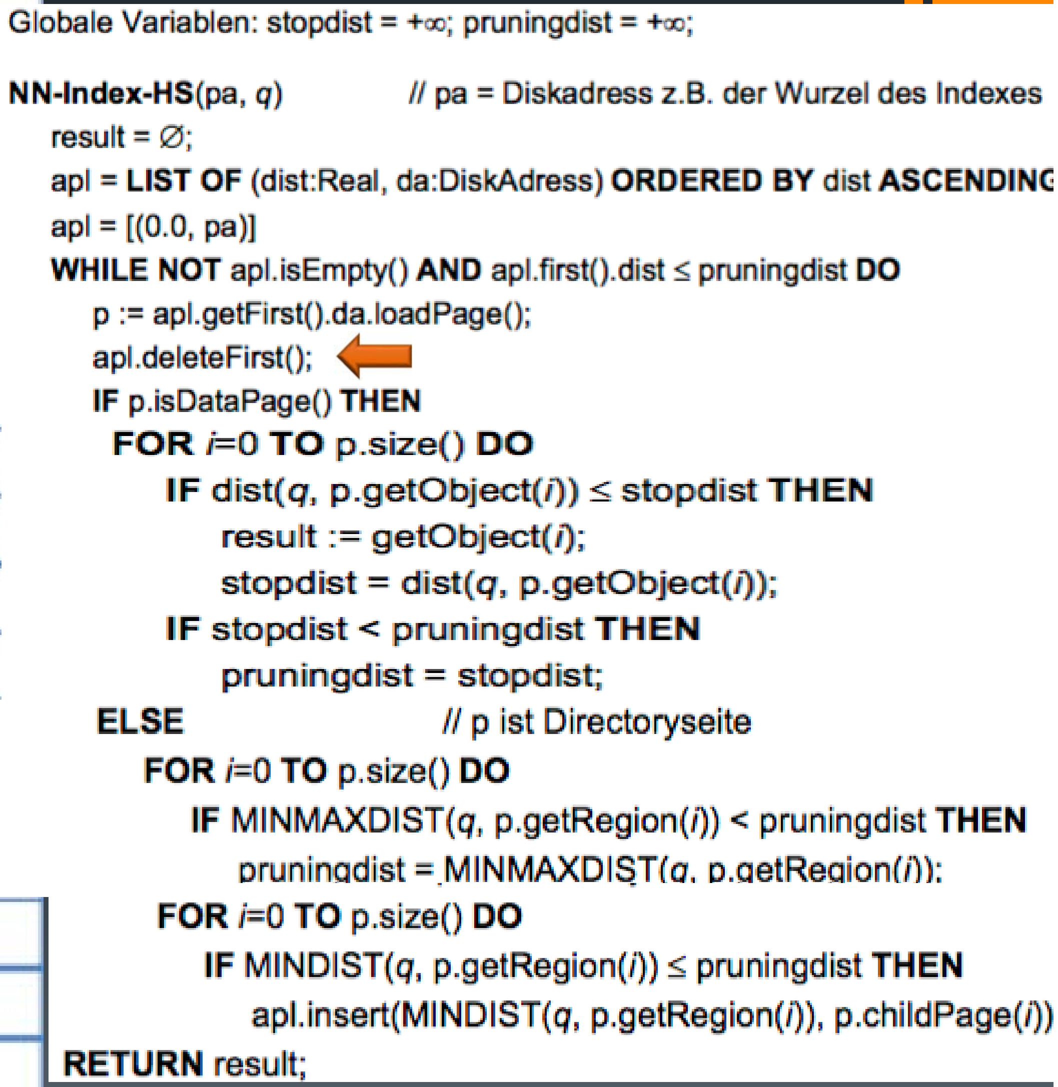
p = p13

APL= [(2,p13),(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13

stopdist = +Inf   pruningdist= 5

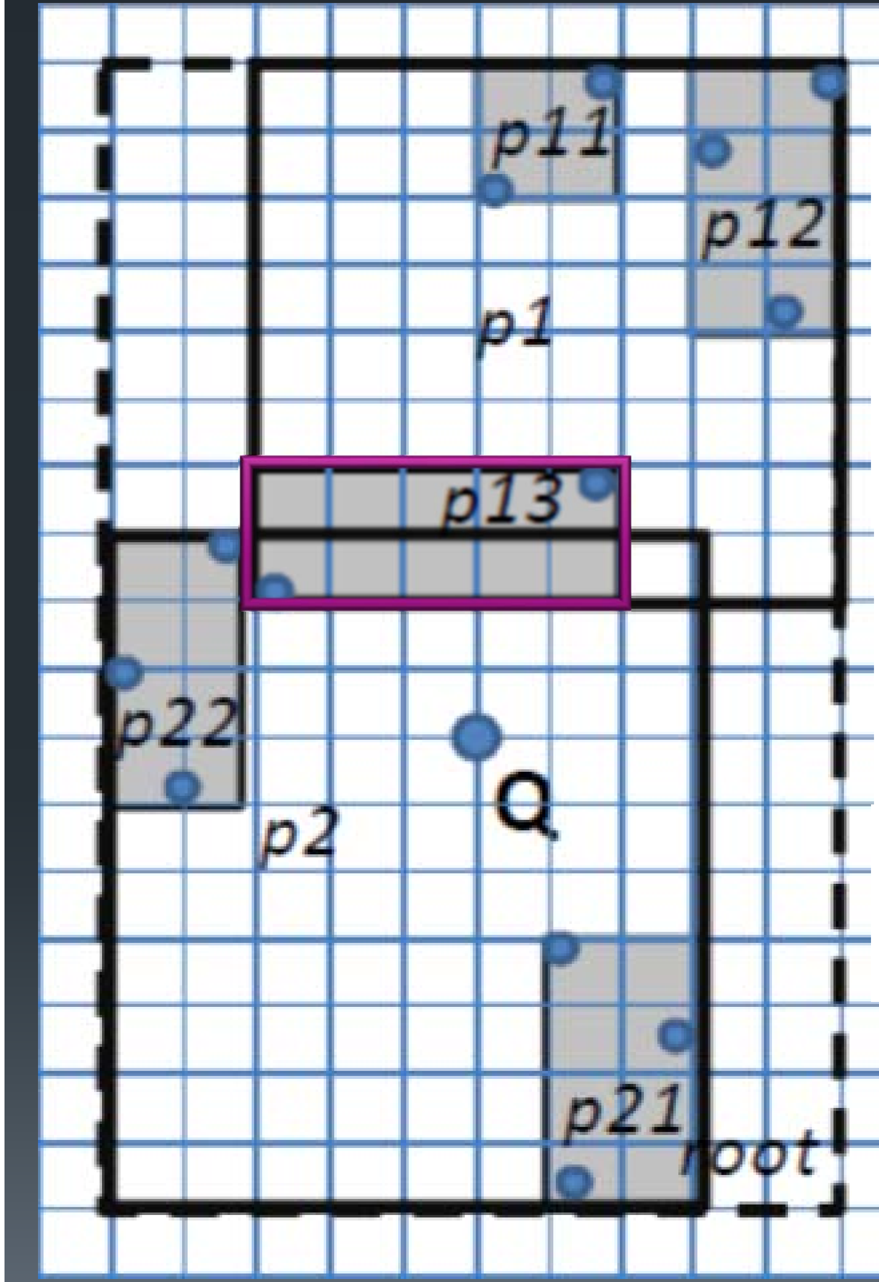




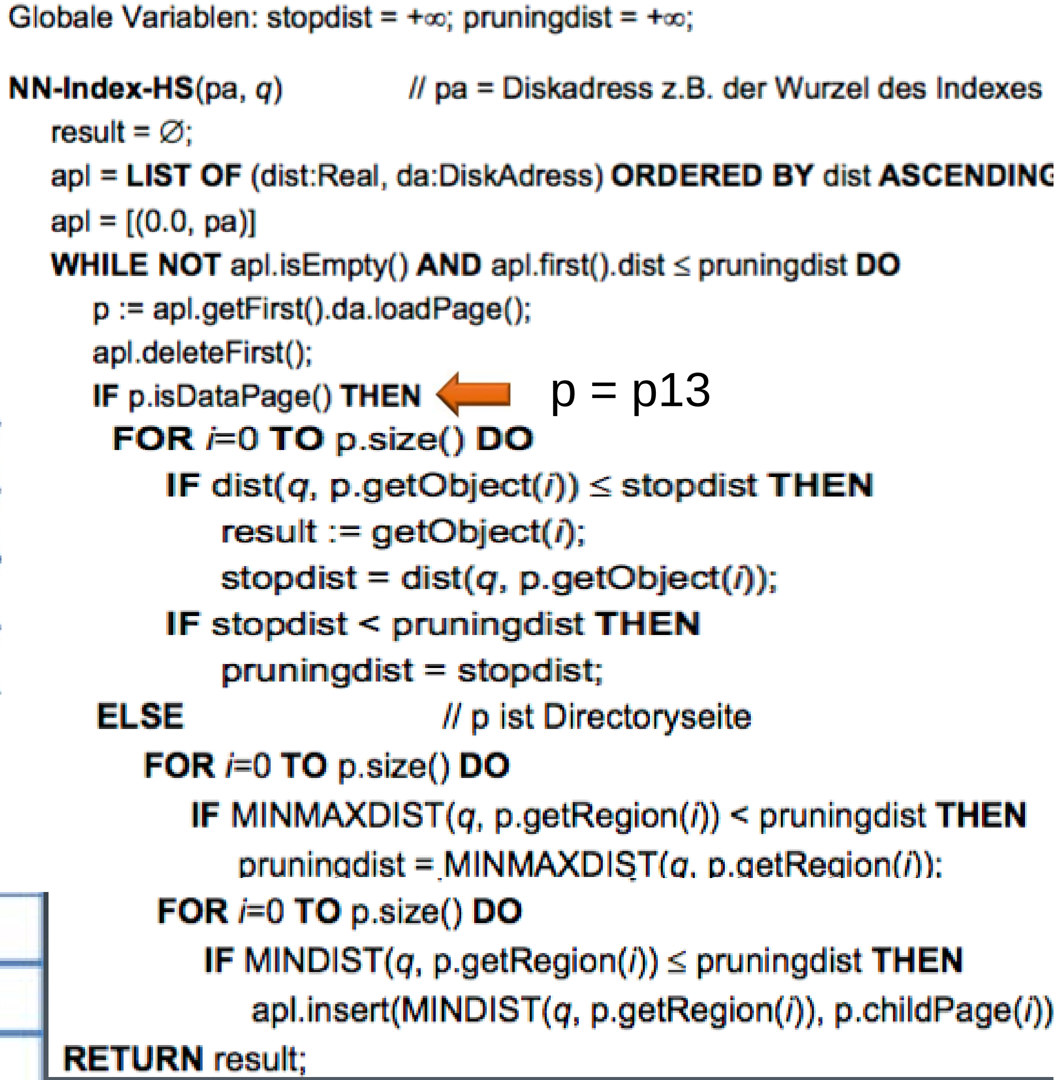
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
        FOR i=0 TO p.size() DO
           IF dist(q, p.getObject(i)) ≤ stopdist THEN
              result := getObject(i);
              stopdist = dist(q, p.getObject(i));
           IF stopdist < pruningdist THEN
              pruningdist = stopdist;
      ELSE                        // p ist Directoryseite
         FOR i=0 TO p.size() DO
            IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
               pruningdist = MINMAXDIST(q, p.getRegion(i));
         FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
               apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = +Inf   pruningdist= 5

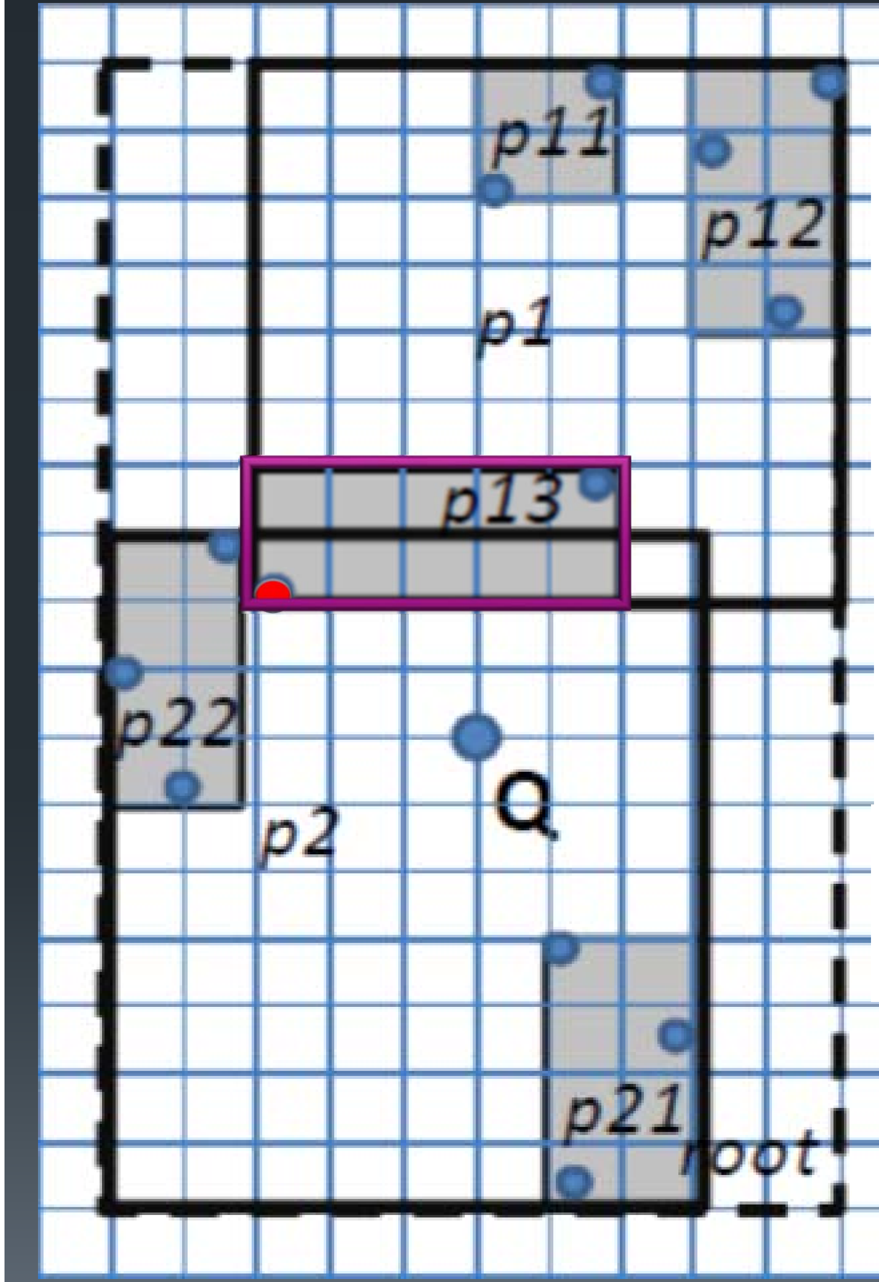


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
       IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
       IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

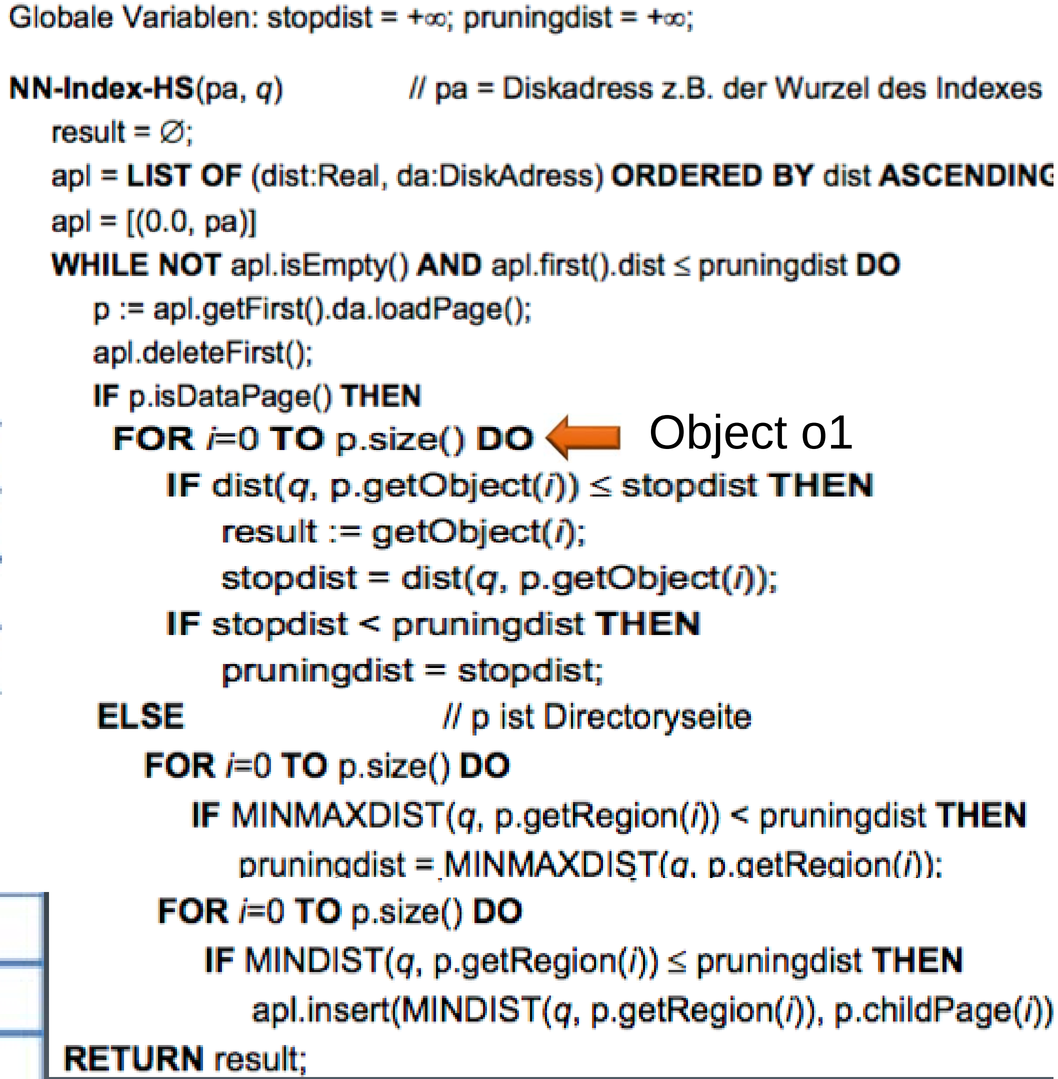
p = p13

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = +Inf pruningdist= 5

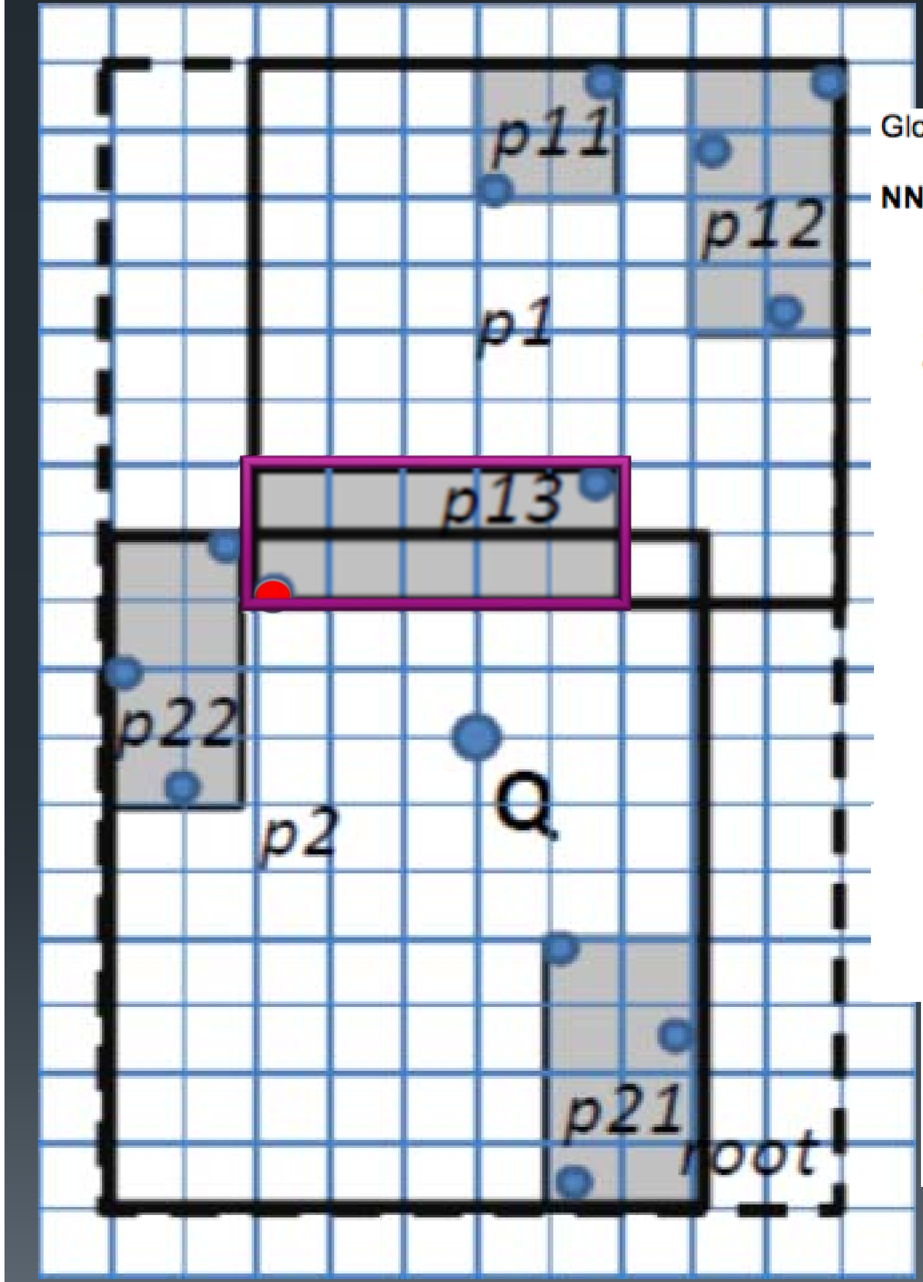


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o1

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = +Inf pruningdist= 5

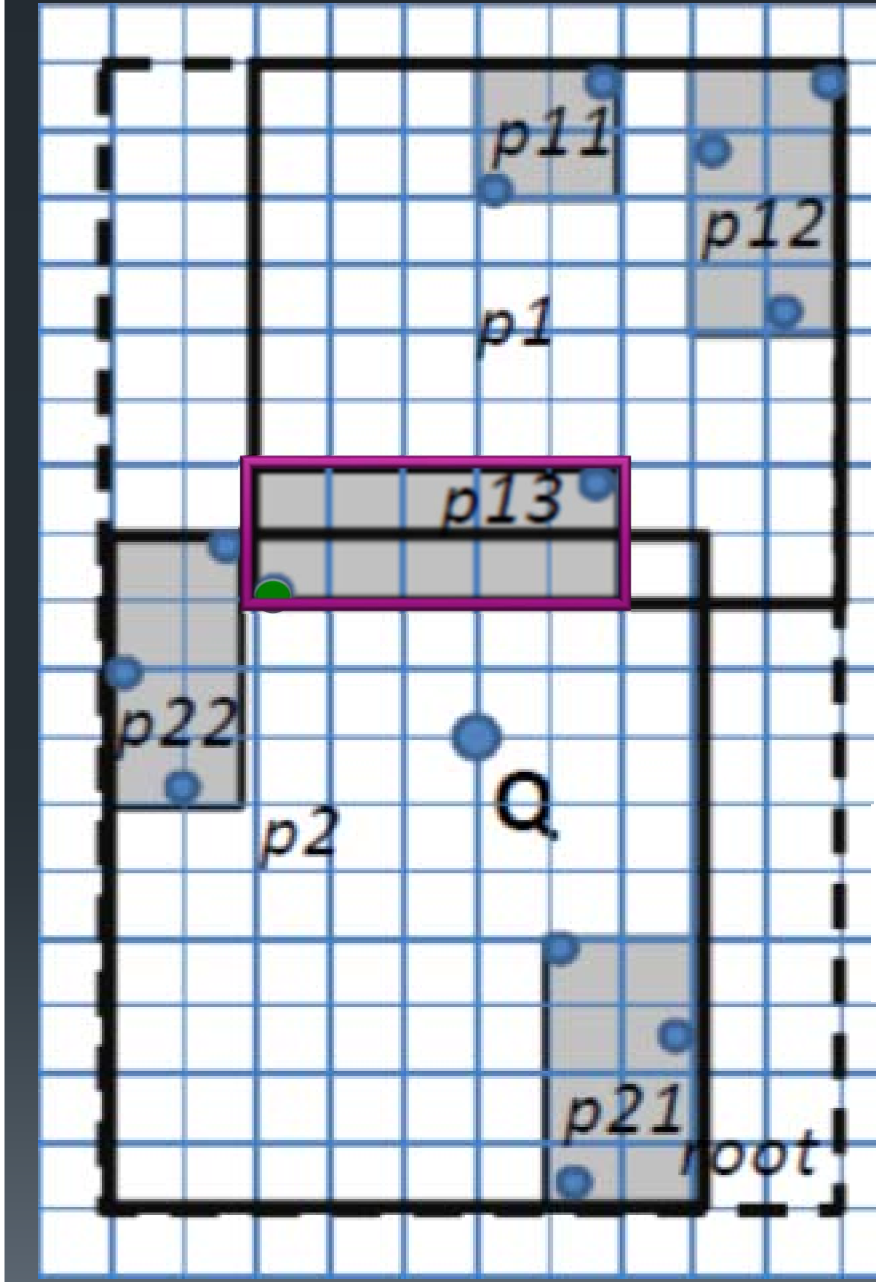
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                            // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

Object o1

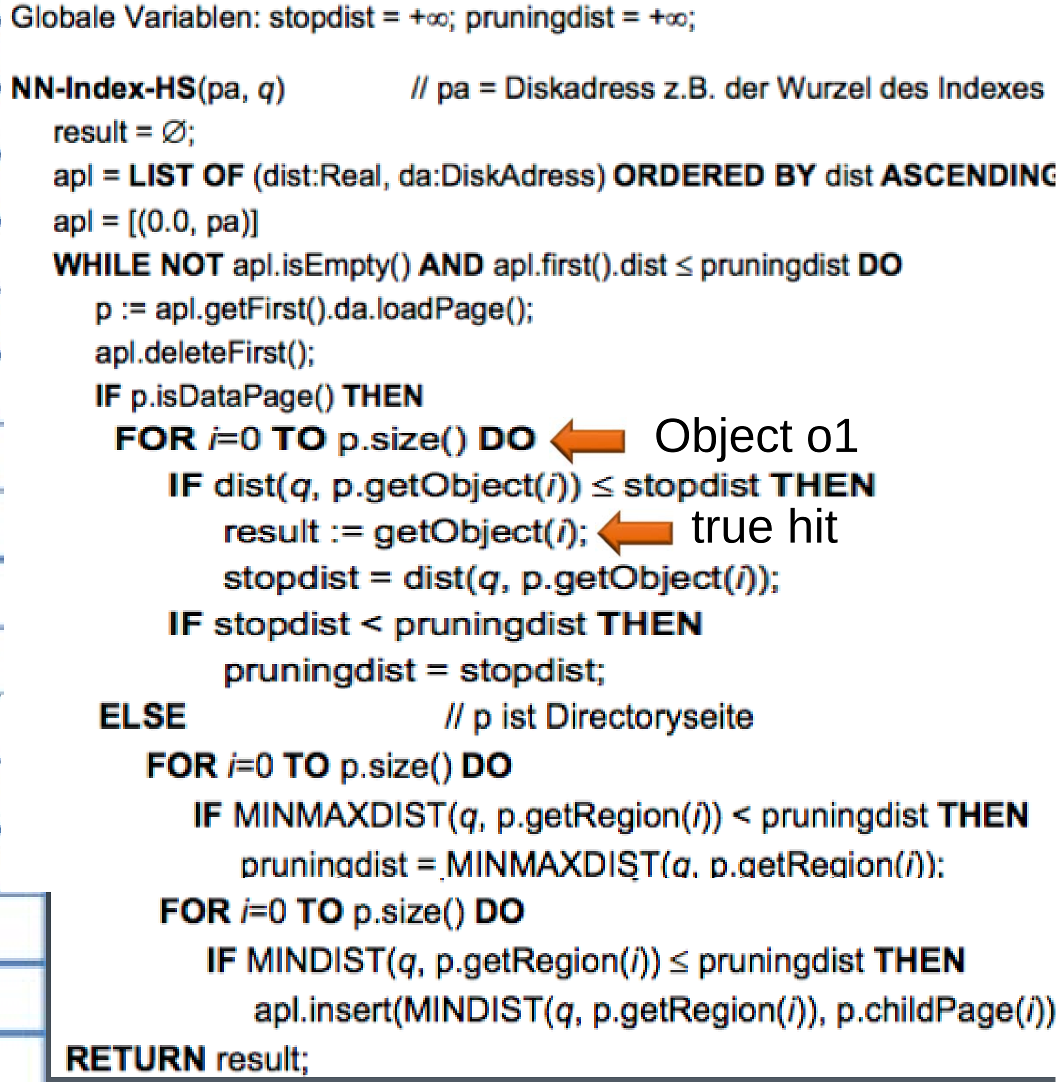
dist(q,o1)=5 <= +Inf

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = +Inf pruningdist= 5



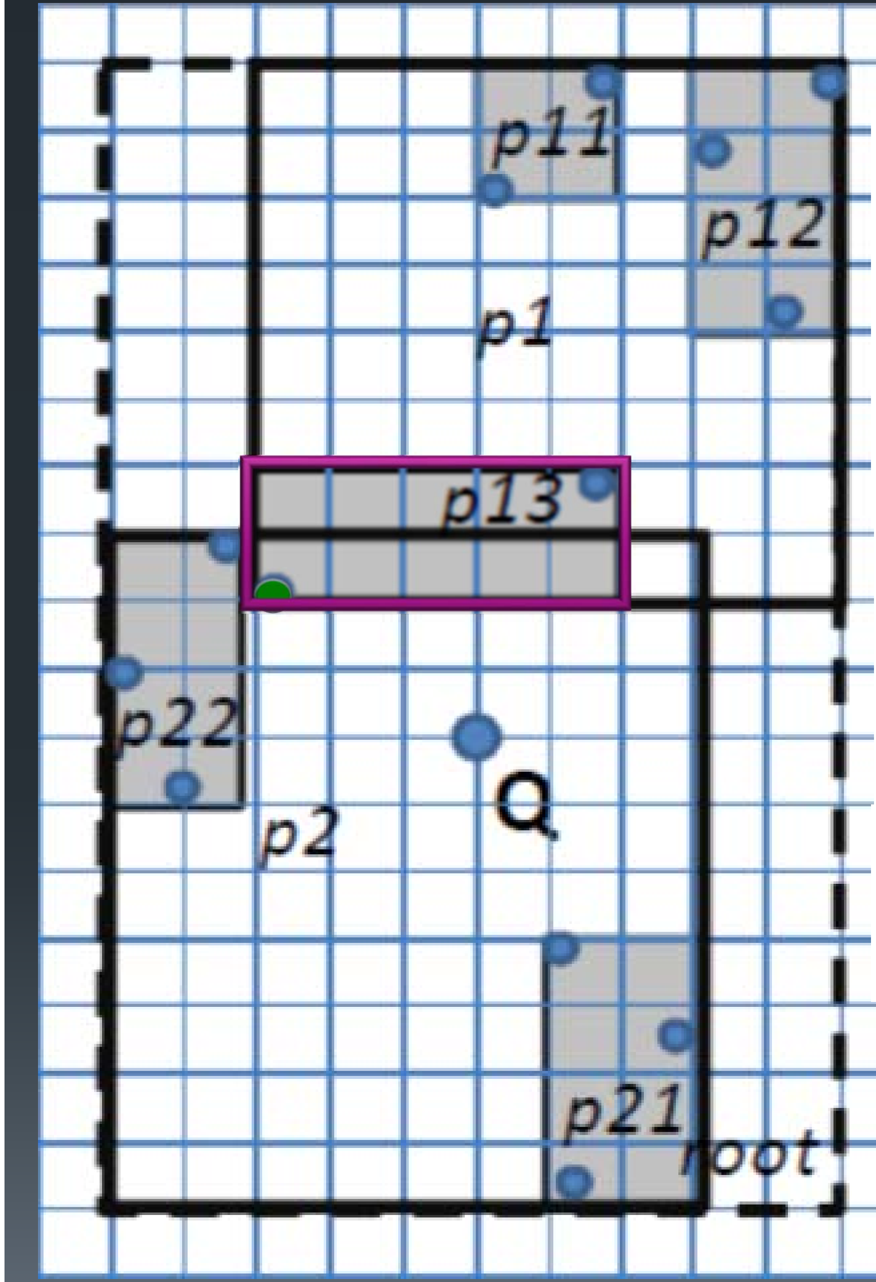
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDINC
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

Object o1

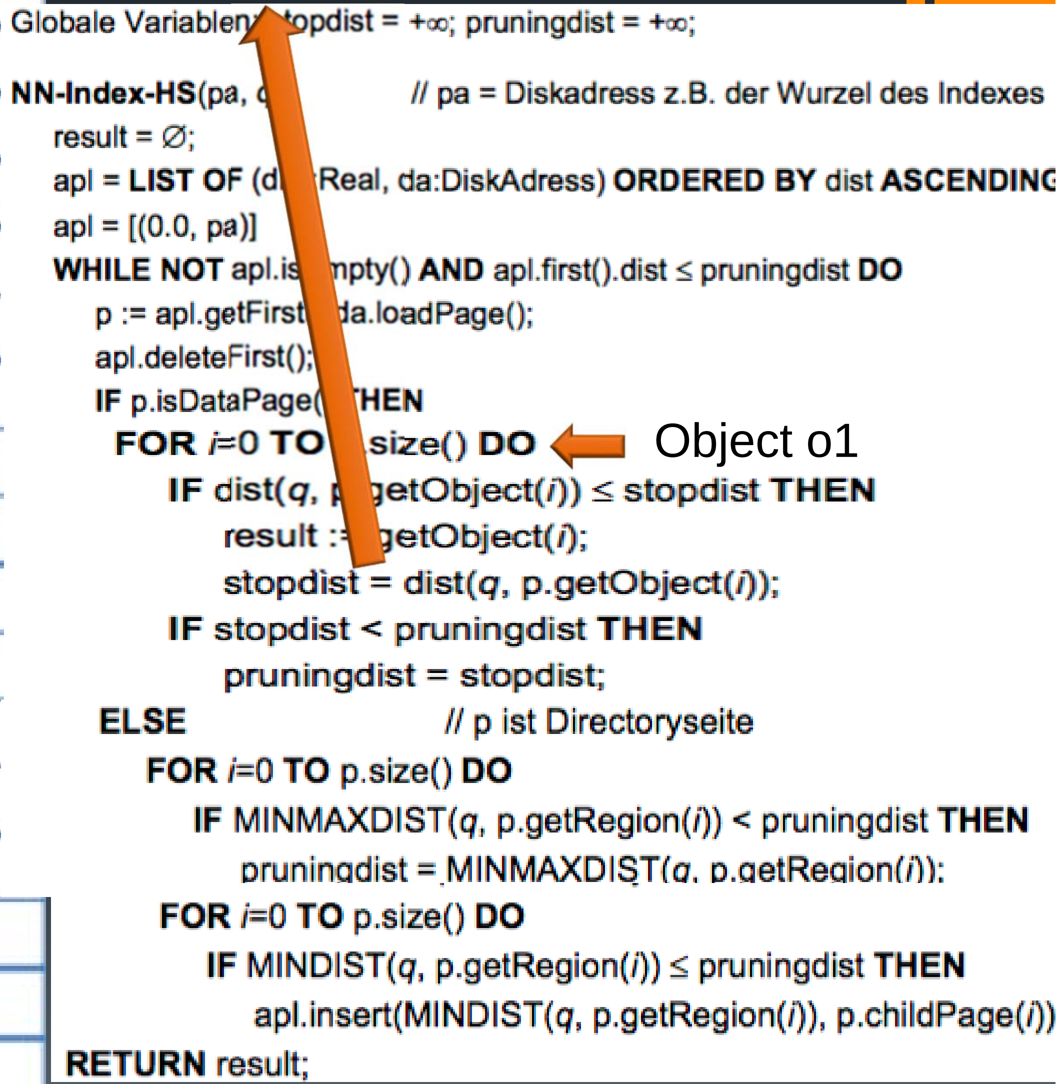
true hit

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = 5 pruningdist= 5

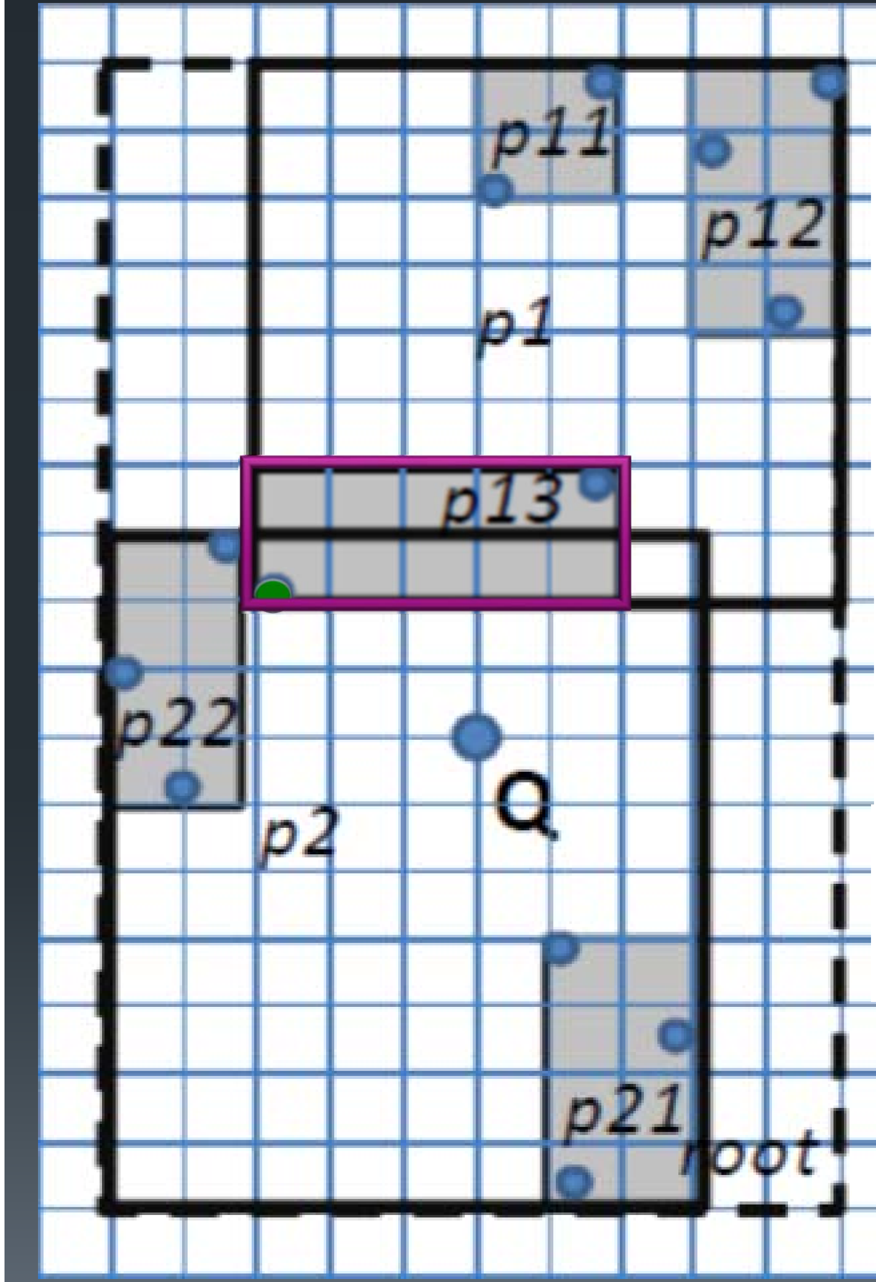


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := p.getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

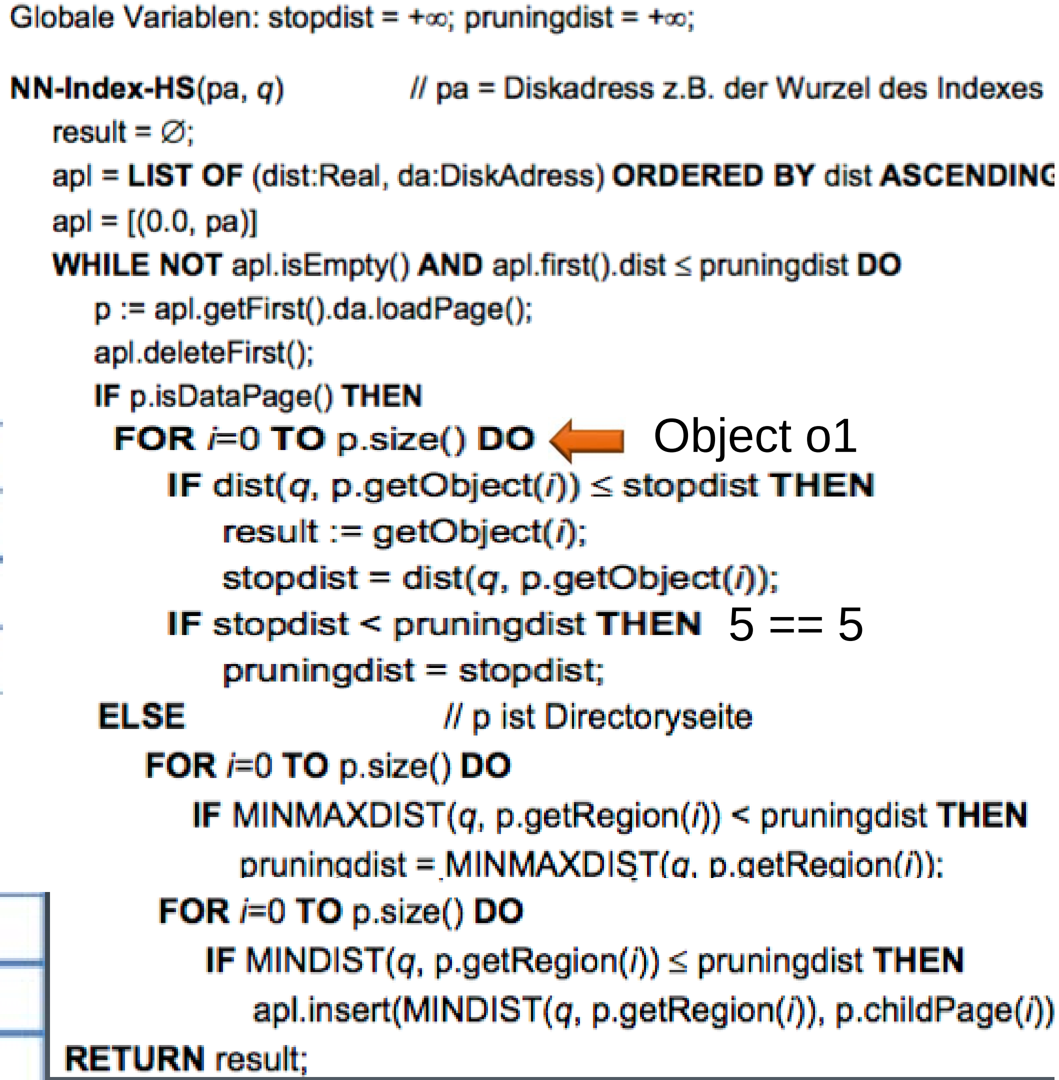
Object o1

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = 5 pruningdist= 5



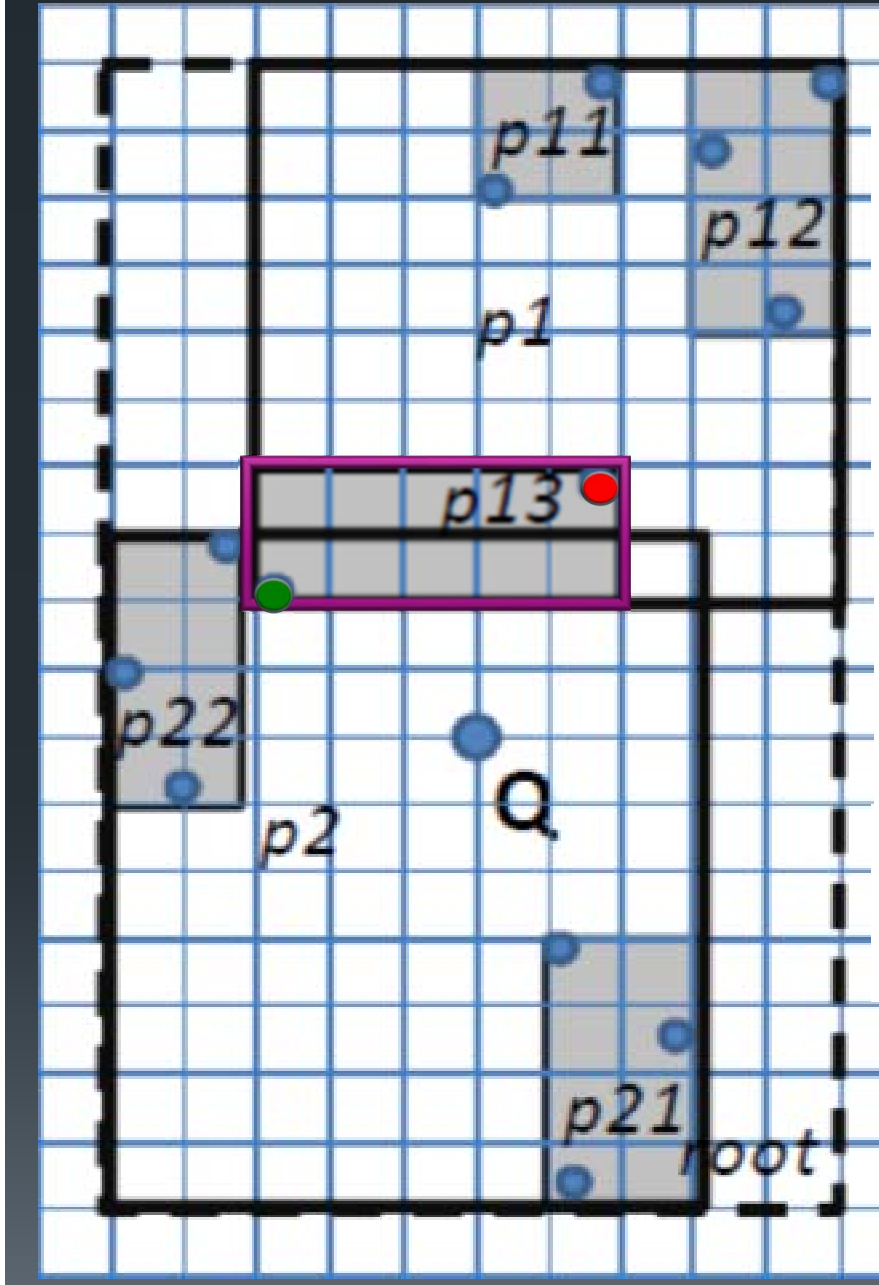
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
   result = ∅;
   apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
   apl = [(0.0, pa)]
   WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
      p := apl.getFirst().da.loadPage();
      apl.deleteFirst();
      IF p.isDataPage() THEN
         FOR i=0 TO p.size() DO
            IF dist(q, p.getObject(i)) ≤ stopdist THEN
               result := getObject(i);
               stopdist = dist(q, p.getObject(i));
            IF stopdist < pruningdist THEN
               pruningdist = stopdist;
      ELSE                          // p ist Directoryseite
         FOR i=0 TO p.size() DO
            IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
               pruningdist = MINMAXDIST(q, p.getRegion(i));
         FOR i=0 TO p.size() DO
            IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
               apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
   RETURN result;
```

Object o1

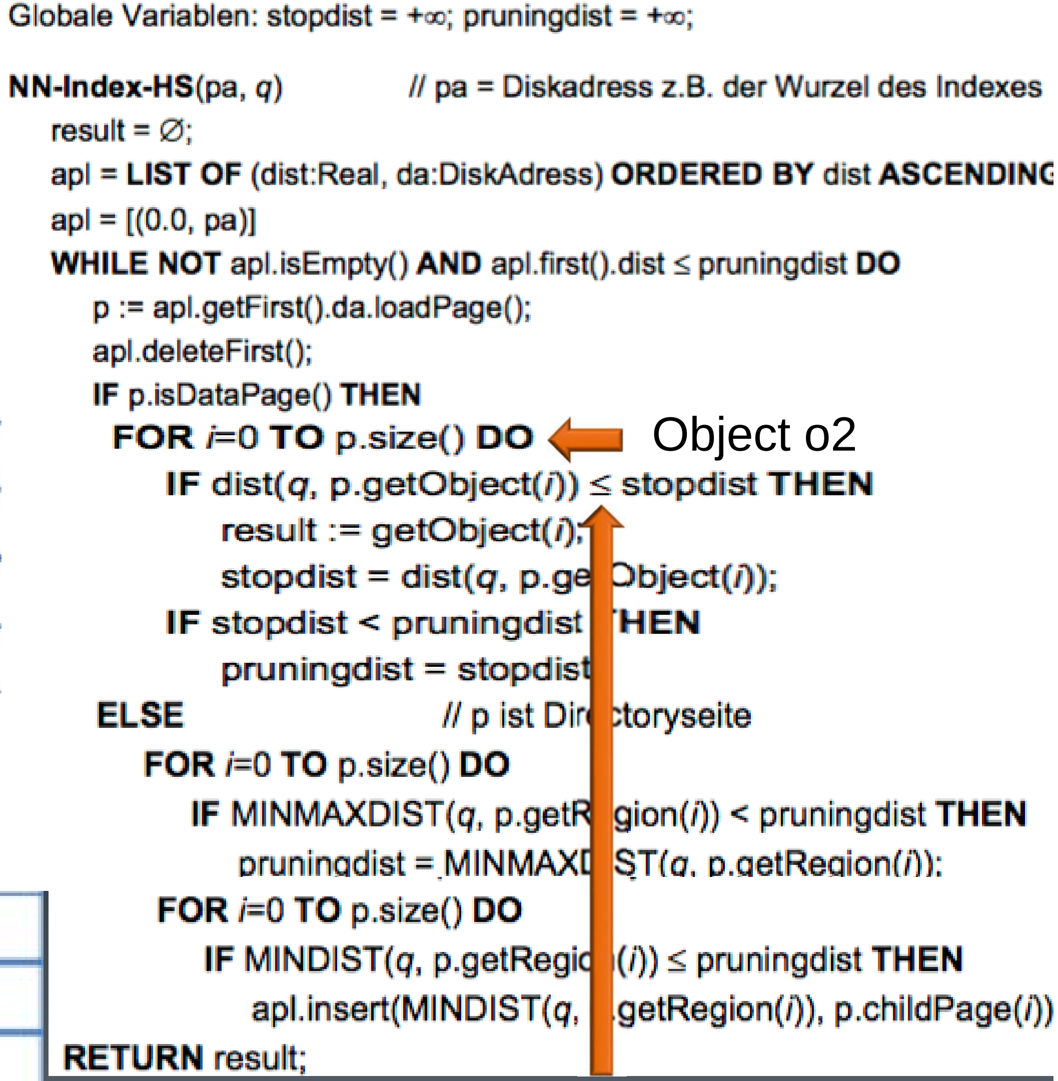
5 == 5

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13



stopdist = 5 pruningdist= 5



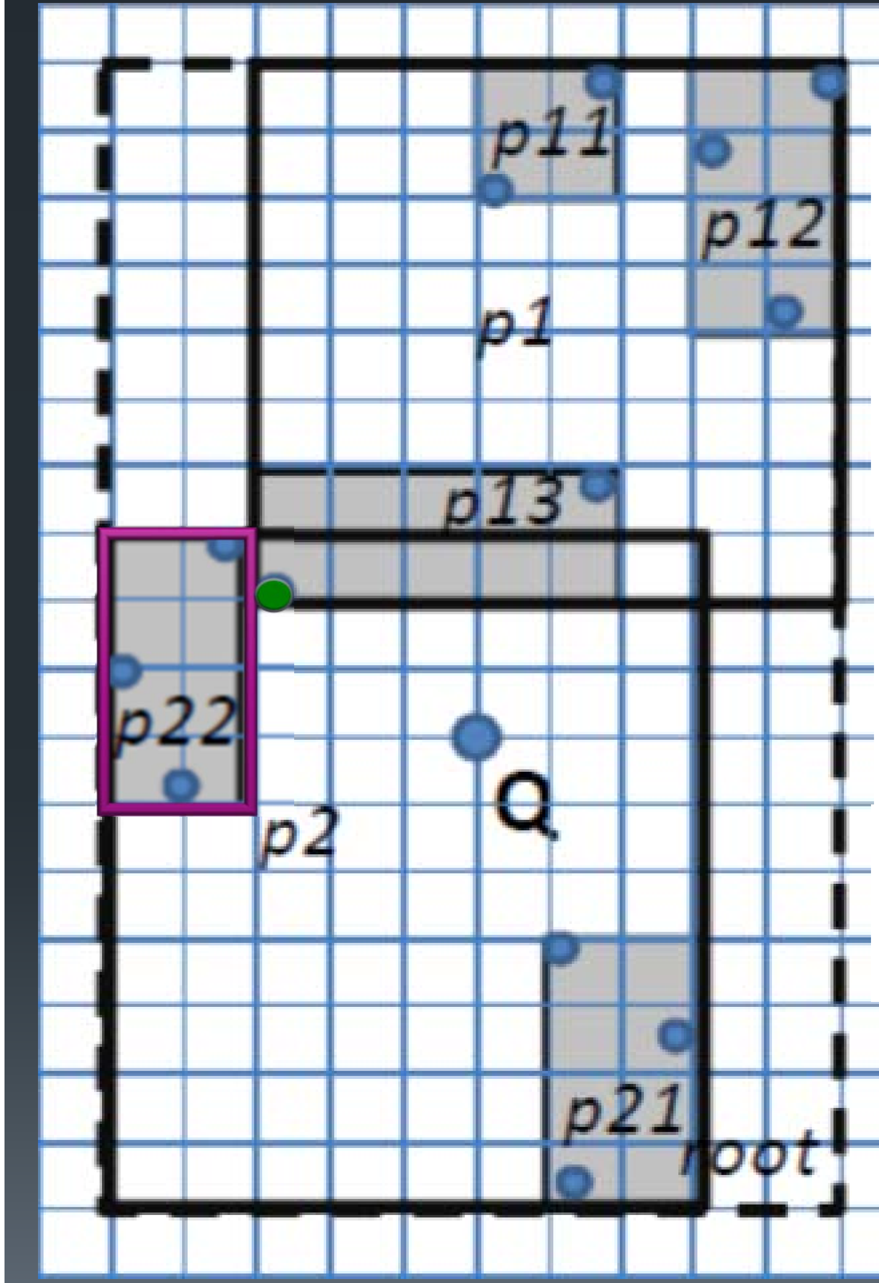
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                    // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

Object o2

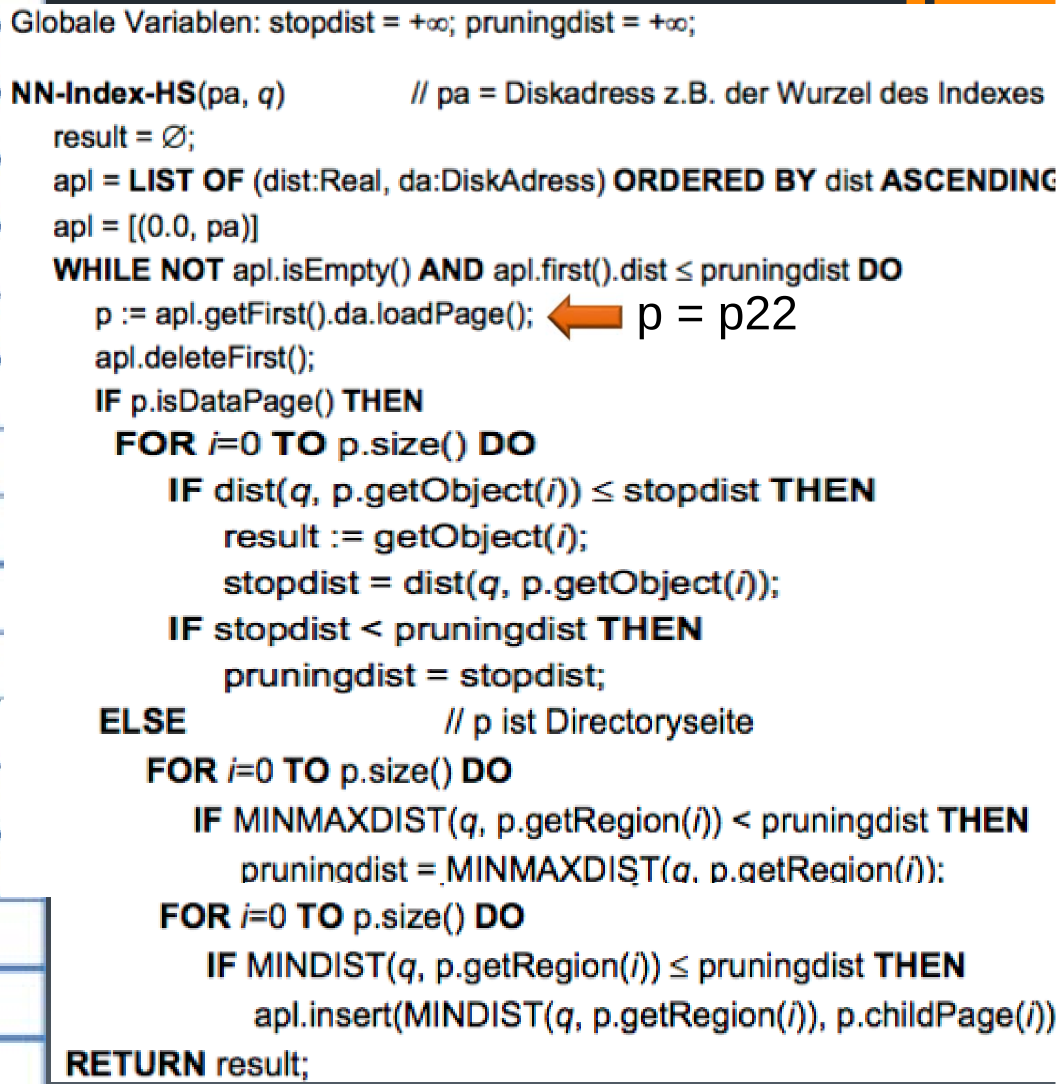
dist(q,o2)=6 <= 5 → Nein, verwirf o2

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



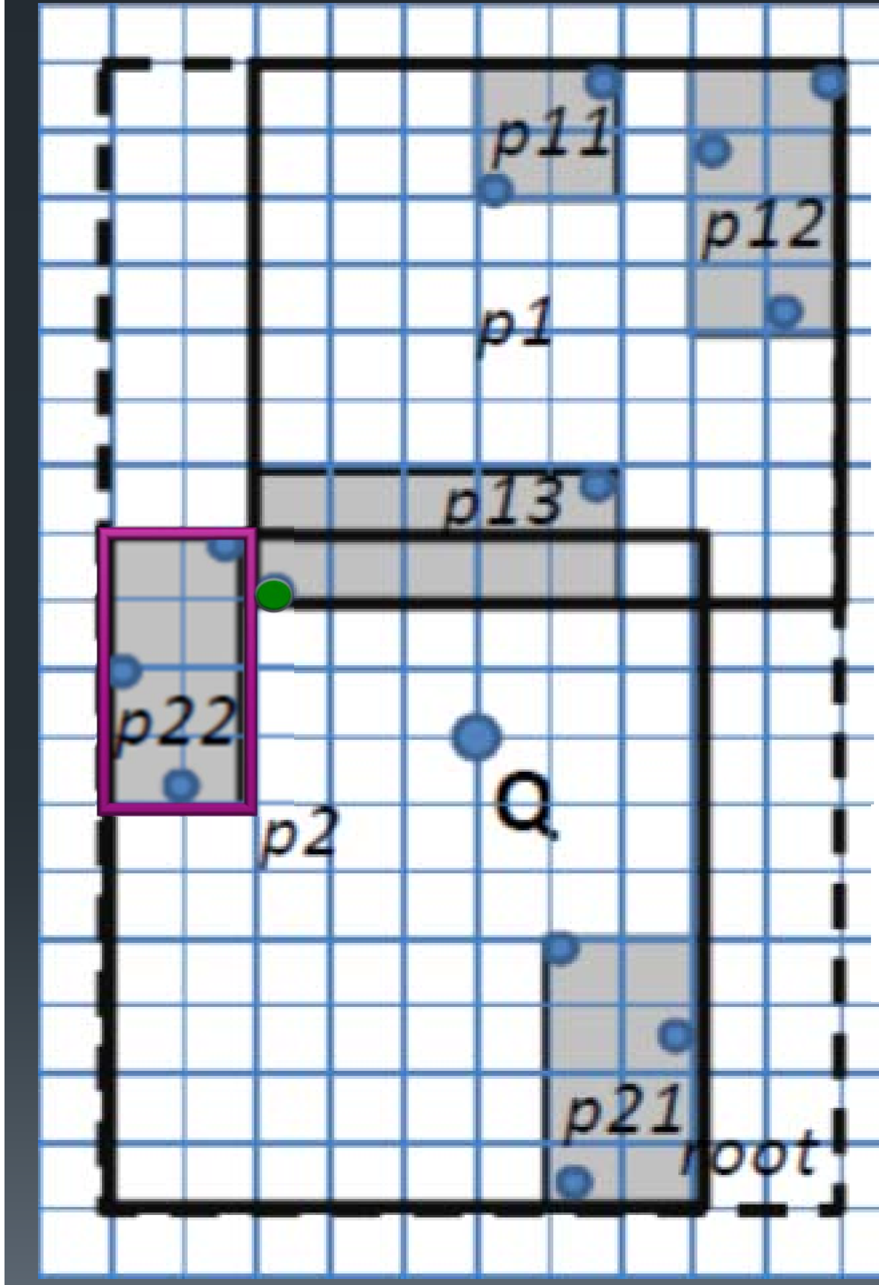
stopdist = 5 pruningdist= 5



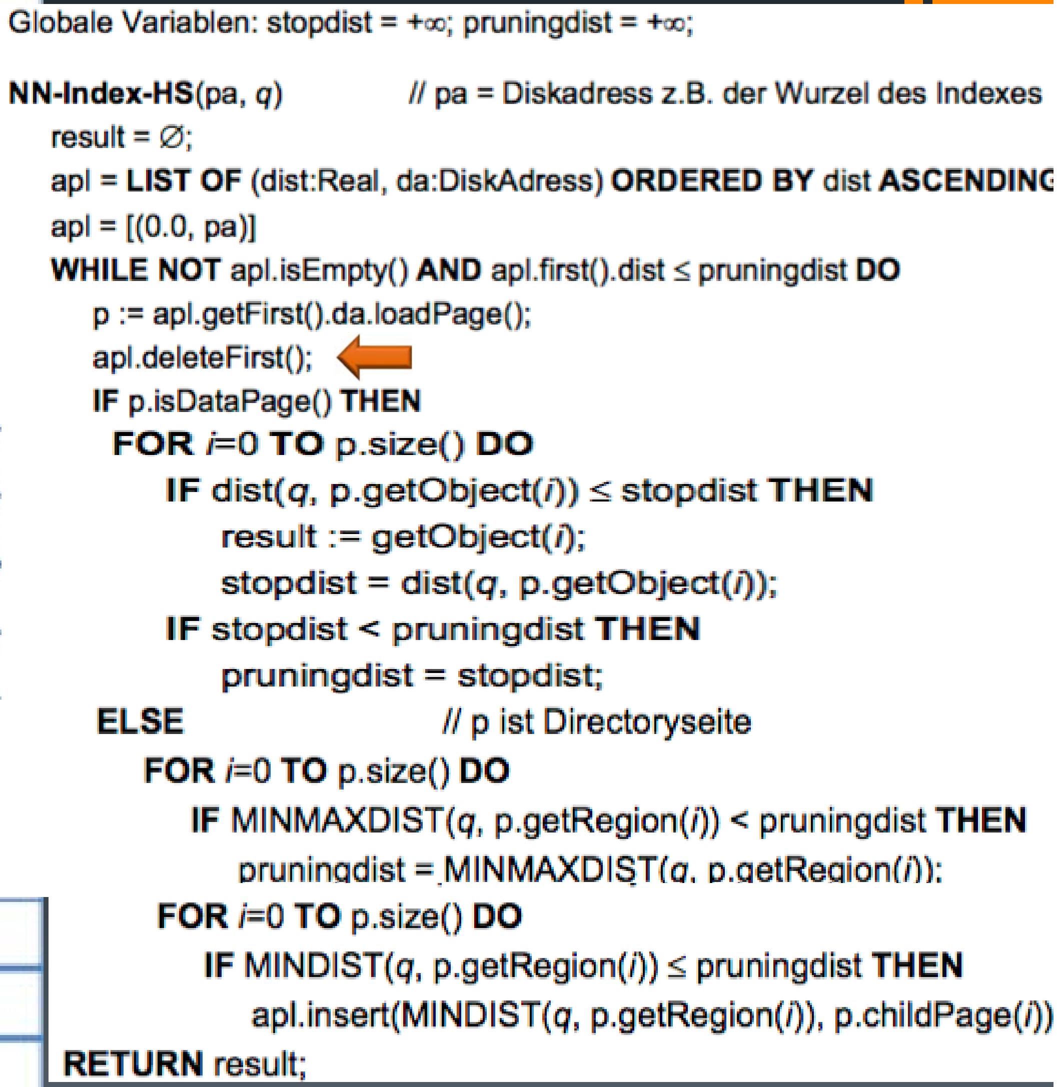
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();   p = p22
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                         // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= [(3,p22),(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



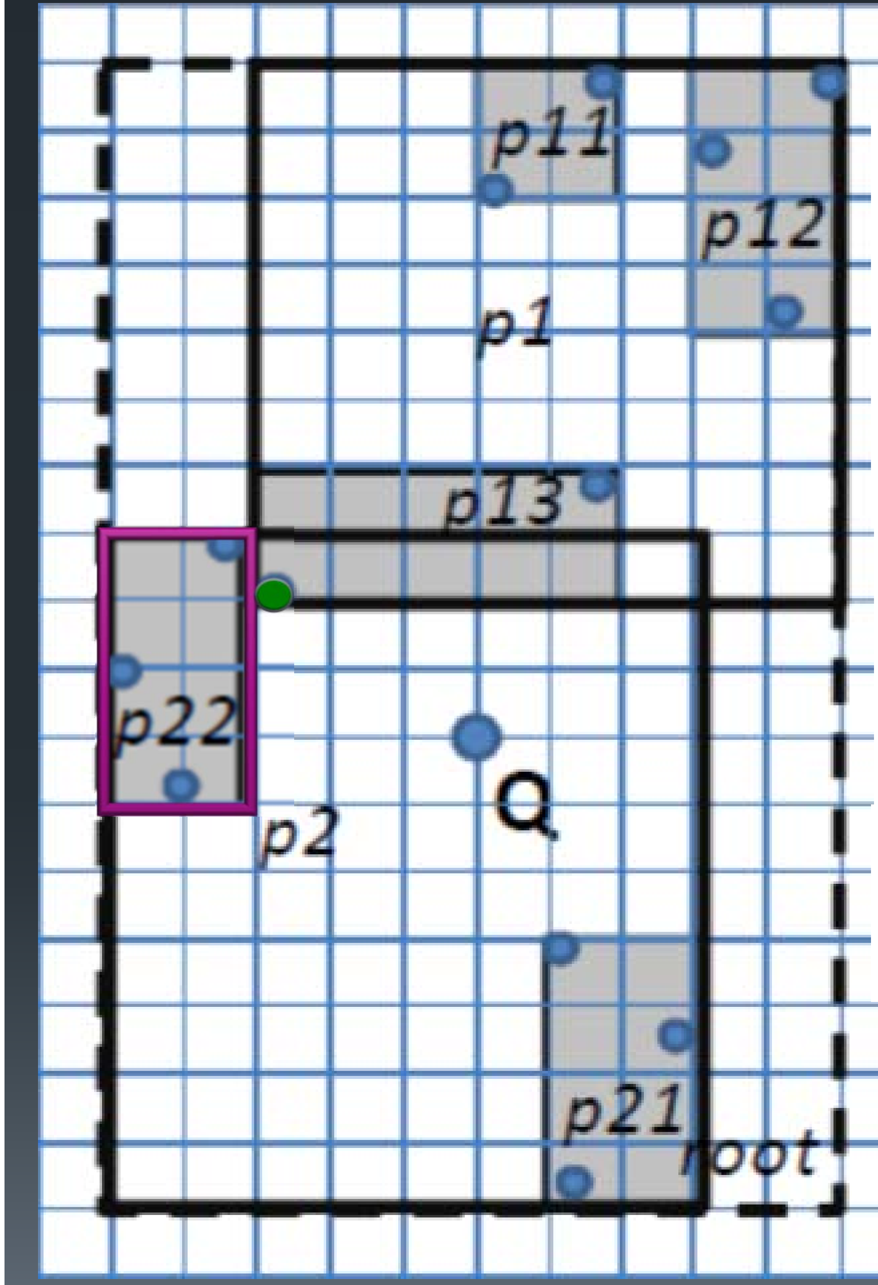
stopdist = 5 pruningdist= 5



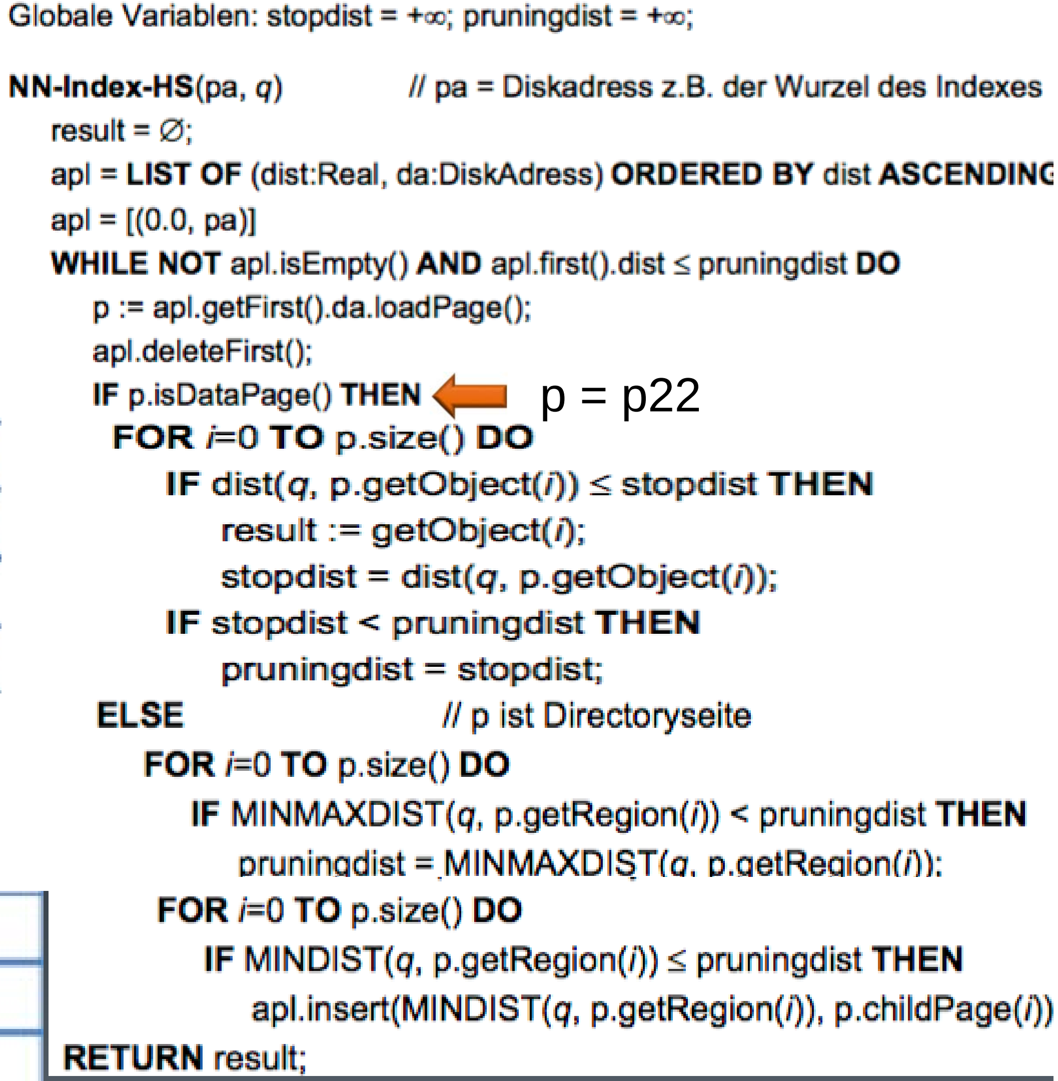
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                    // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5



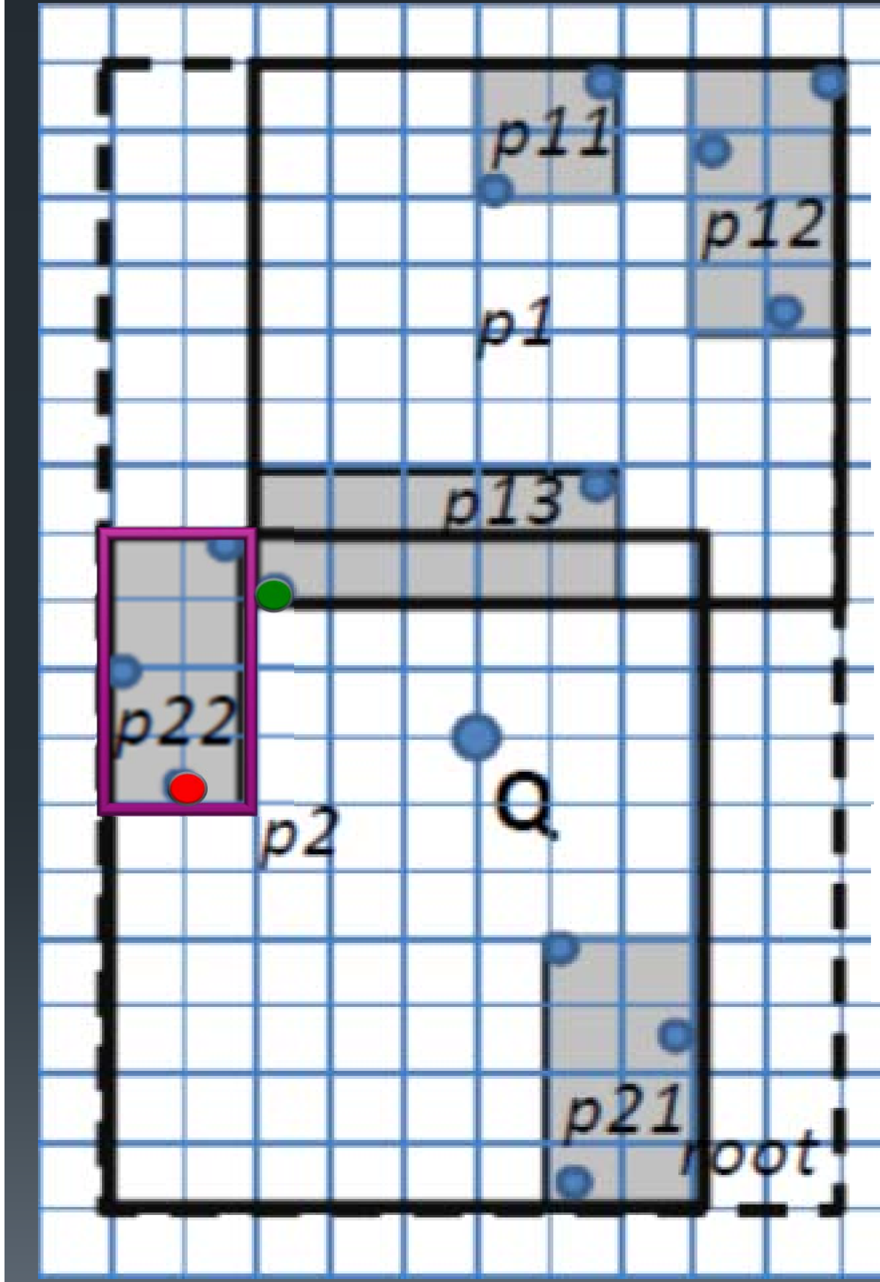
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
           result := getObject(i);
           stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
           pruningdist = stopdist;
    ELSE                            // p ist Directoryseite
      FOR i=0 TO p.size() DO
         IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
            pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
         IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
            apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
 RETURN result;
```

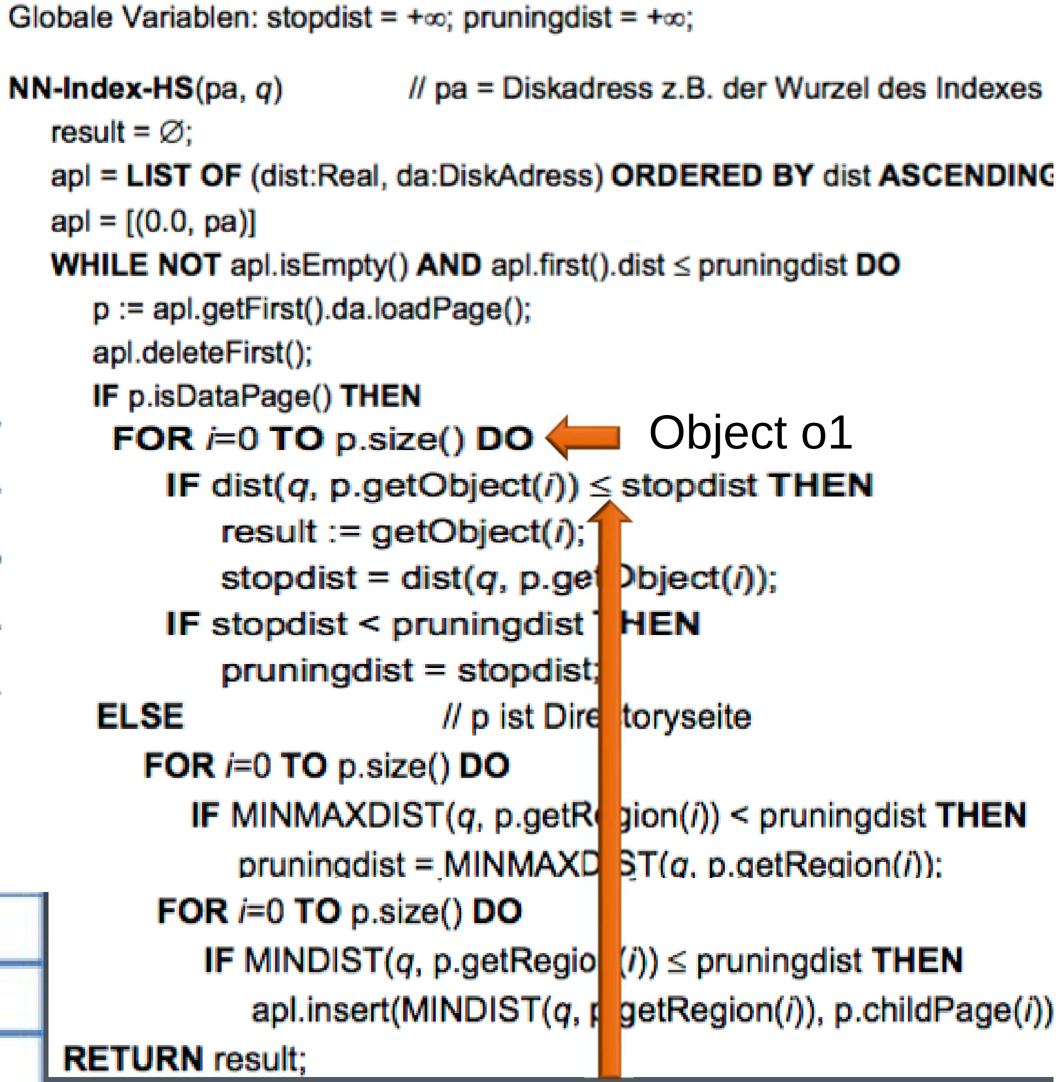
p = p22

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22

stopdist = 5 pruningdist= 5





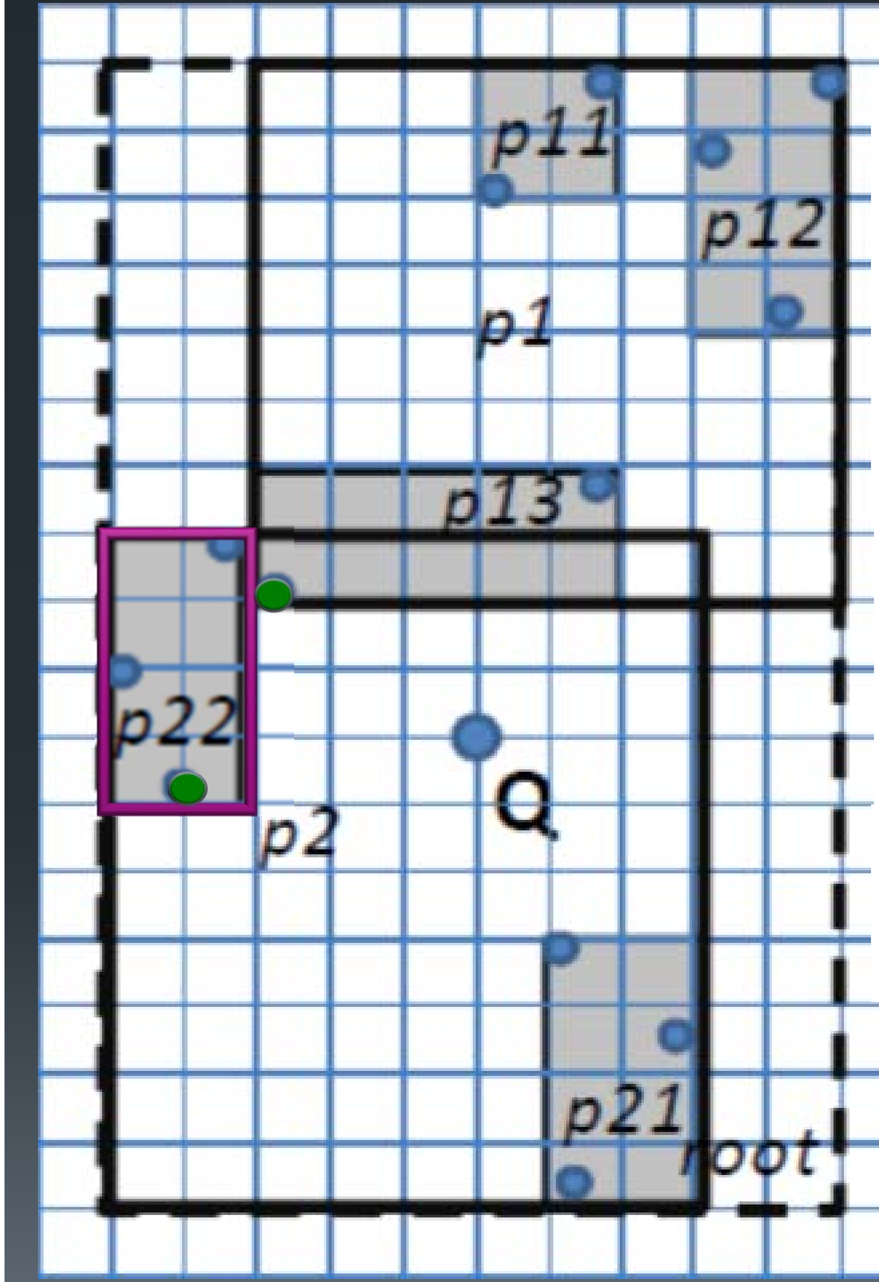
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
           result := getObject(i);
           stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
           pruningdist = stopdist;
    ELSE                           // p ist Directoryseite
      FOR i=0 TO p.size() DO
         IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
           pruningdist = MINMAXDIST(q, p.getRegion(i));
       FOR i=0 TO p.size() DO
          IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
             apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o1

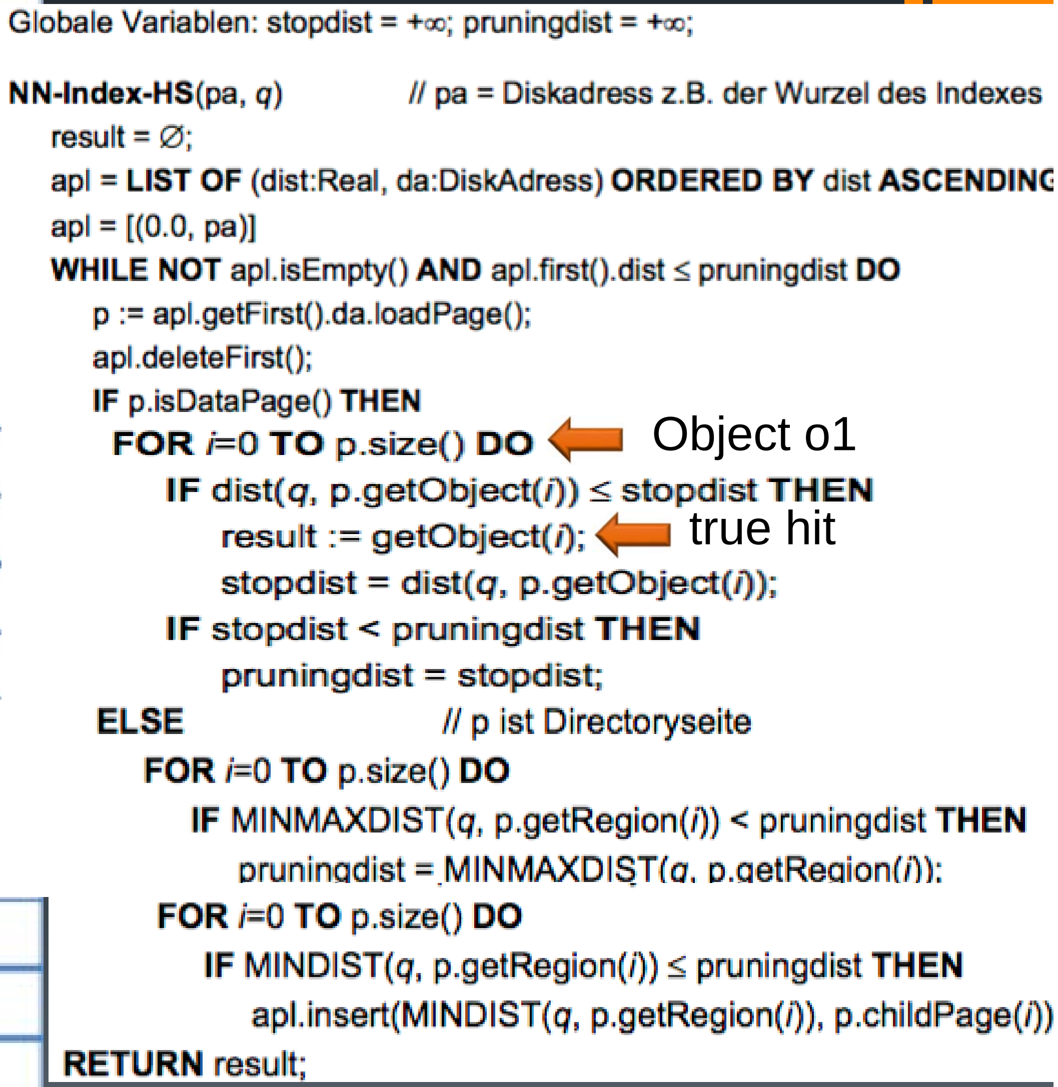
dist(q,o1)=5 <= 5

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5



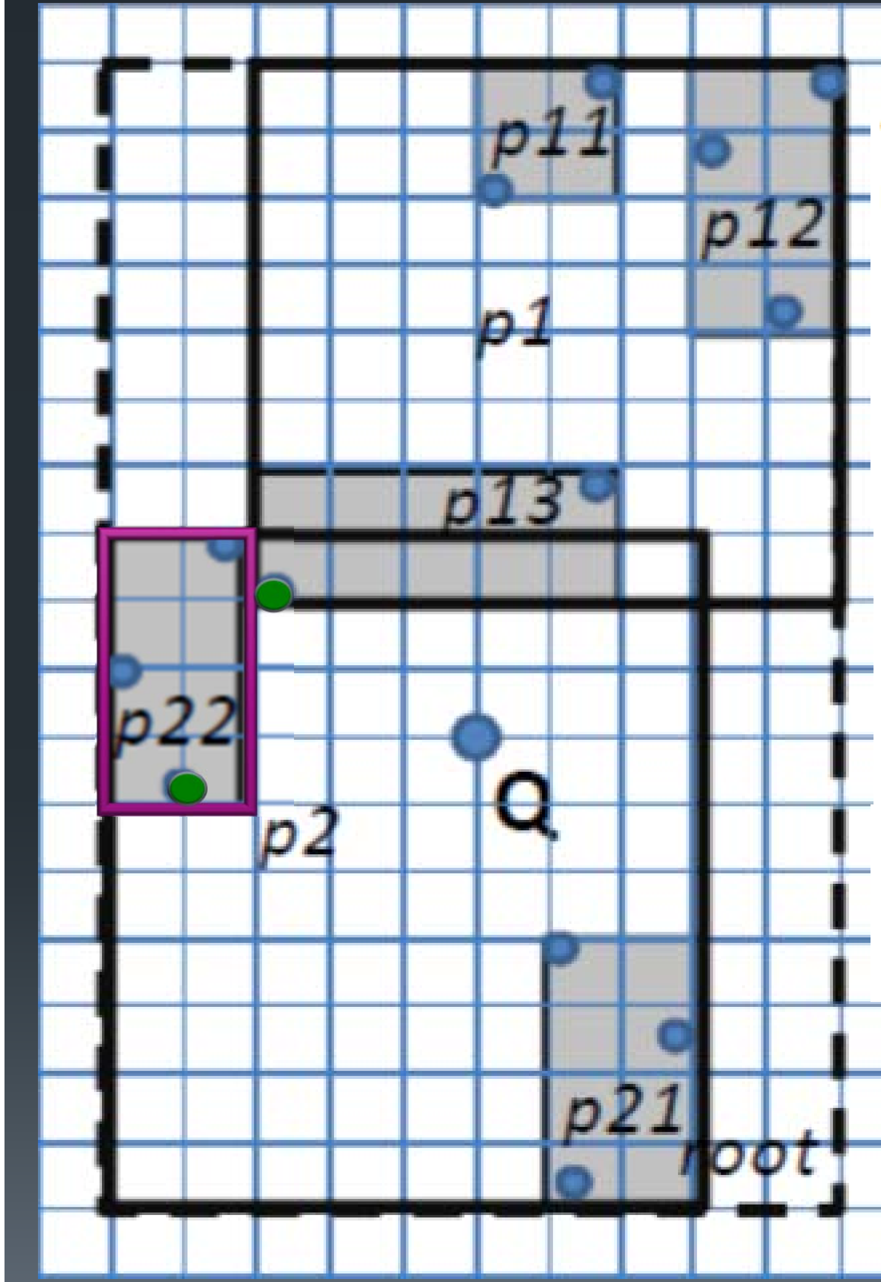
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o1

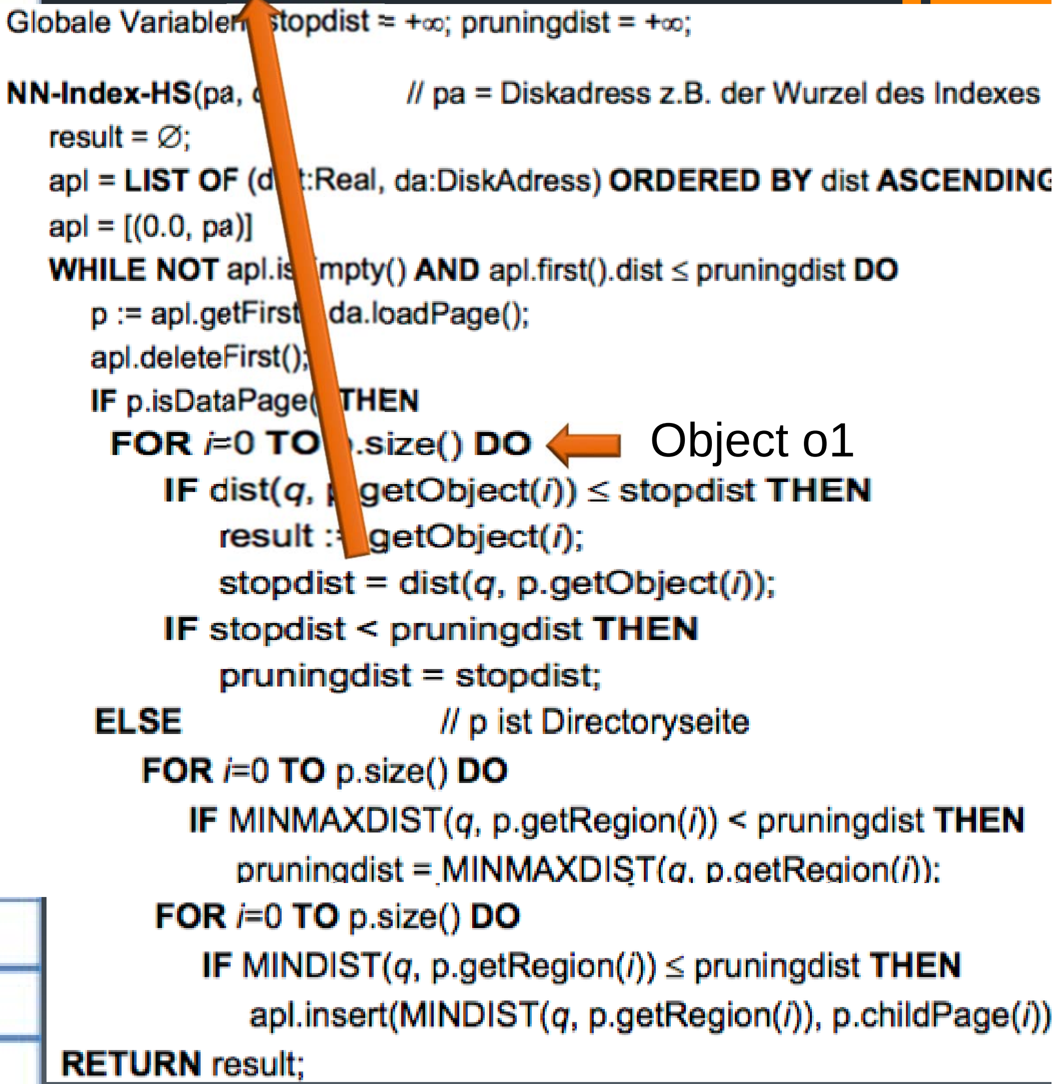
true hit

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5

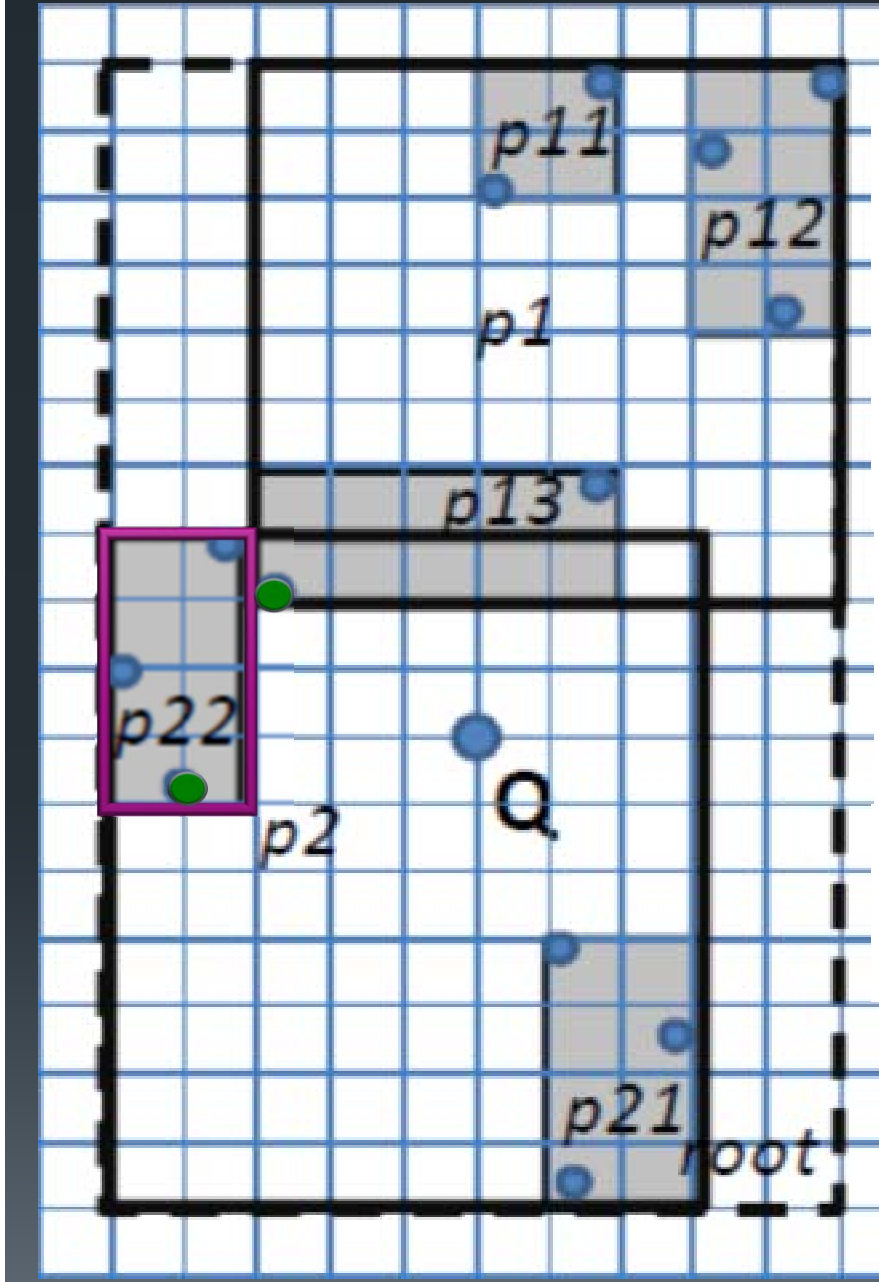


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := p.getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                    // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

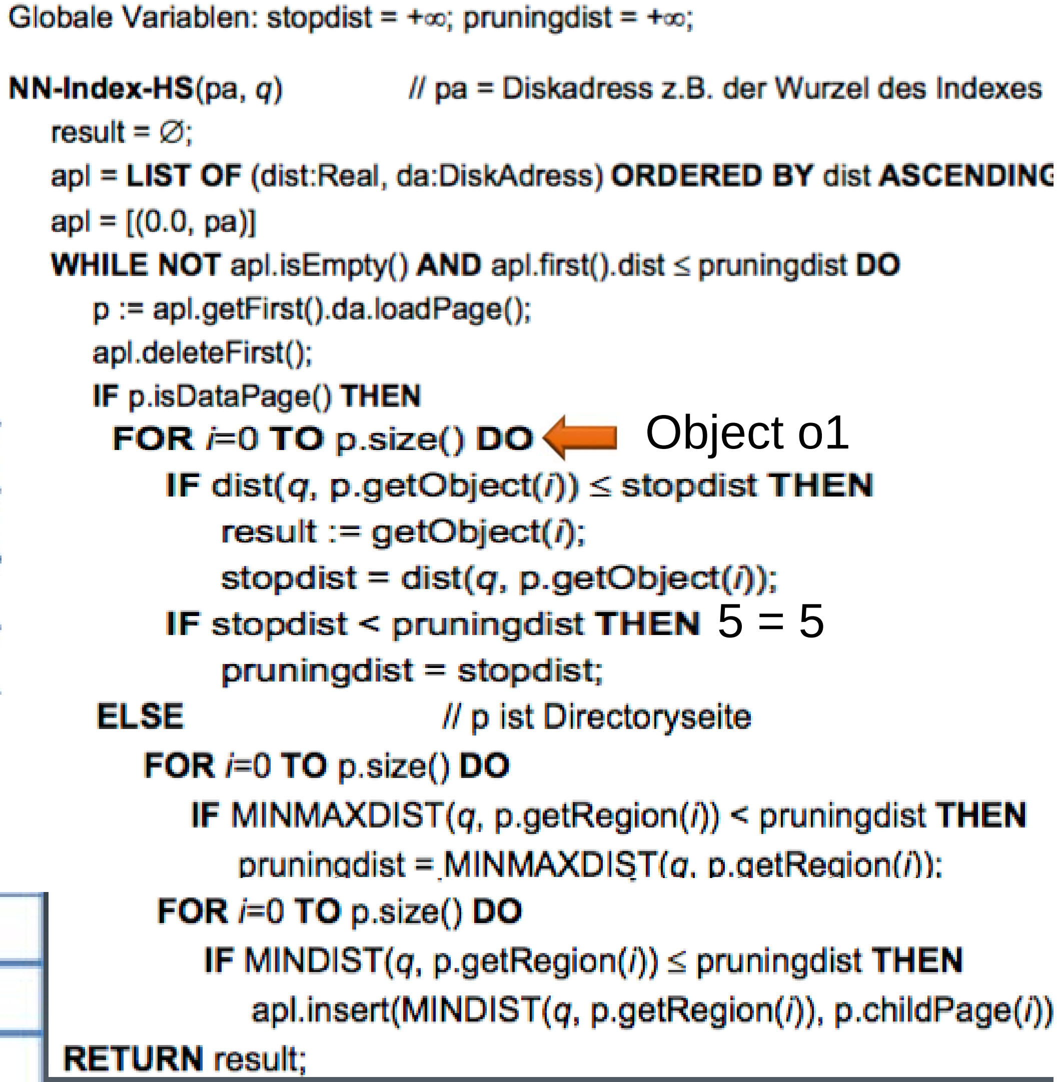
Object o1

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5



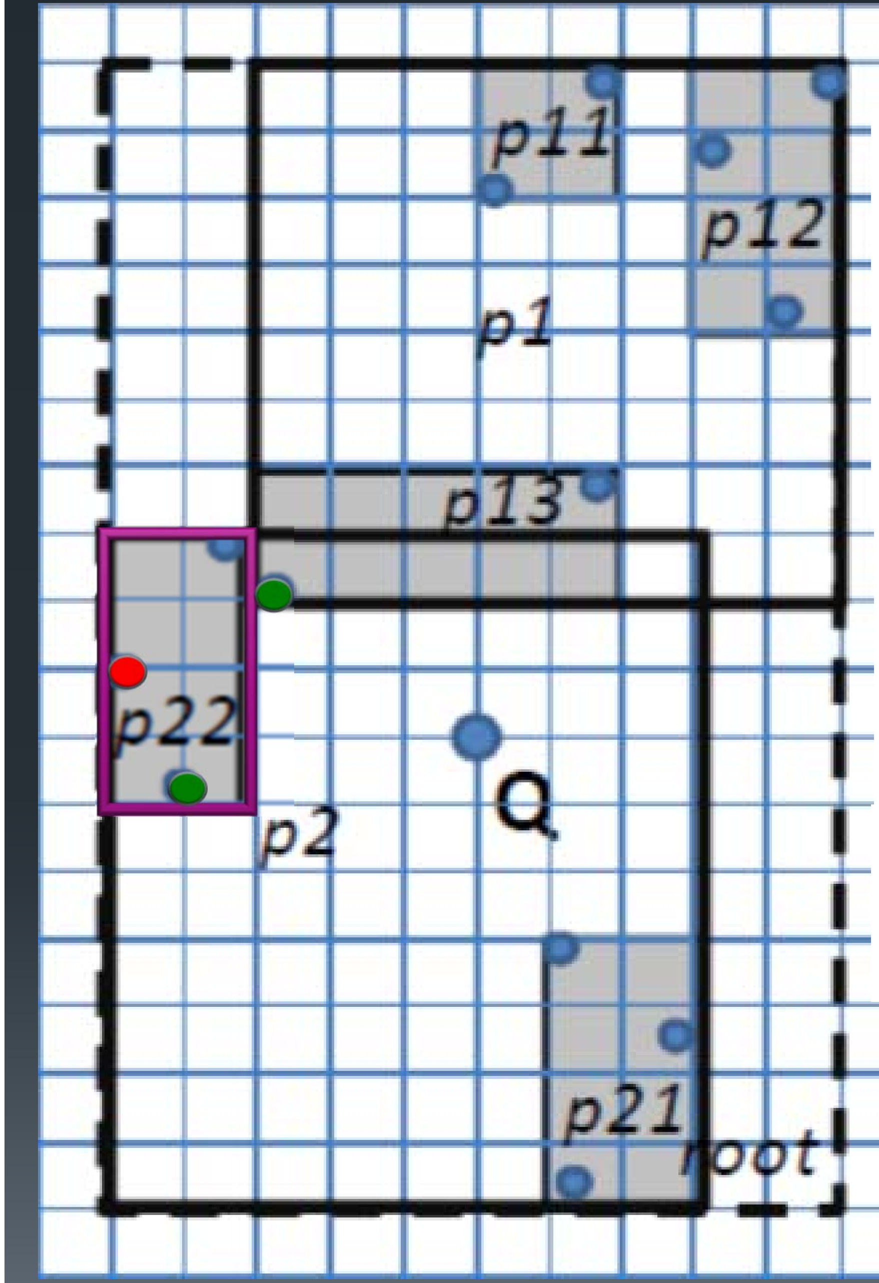
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

Object o1

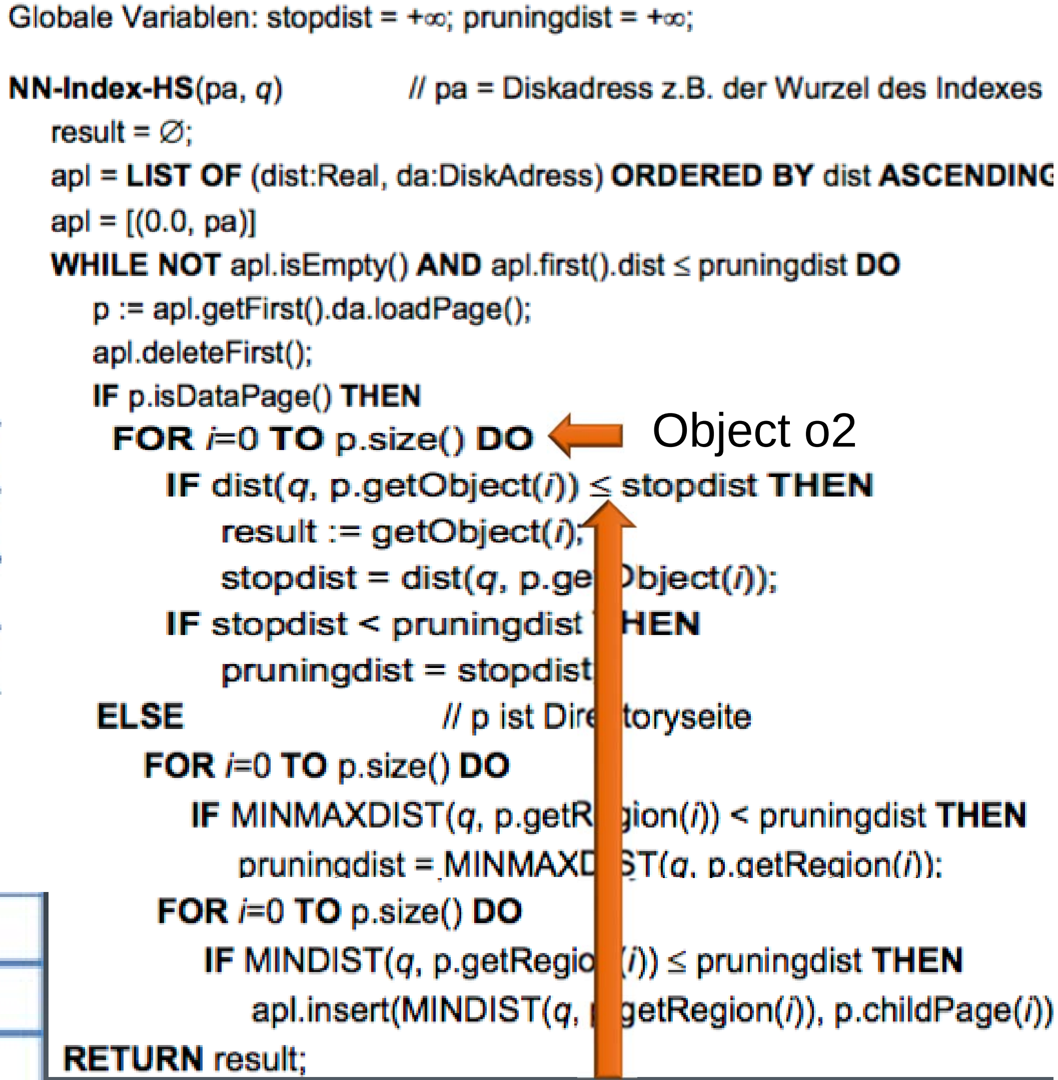
5 = 5

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5



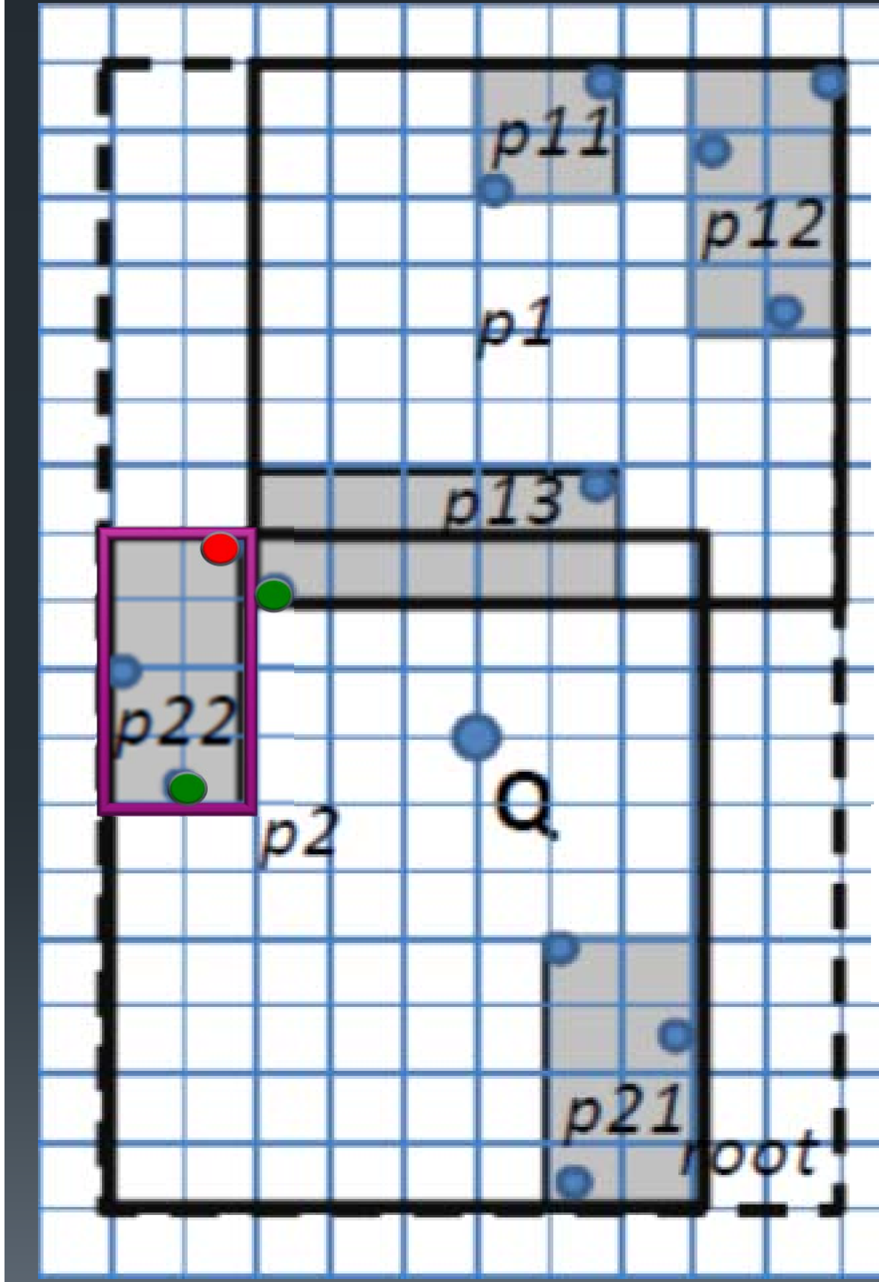
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                         // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o2

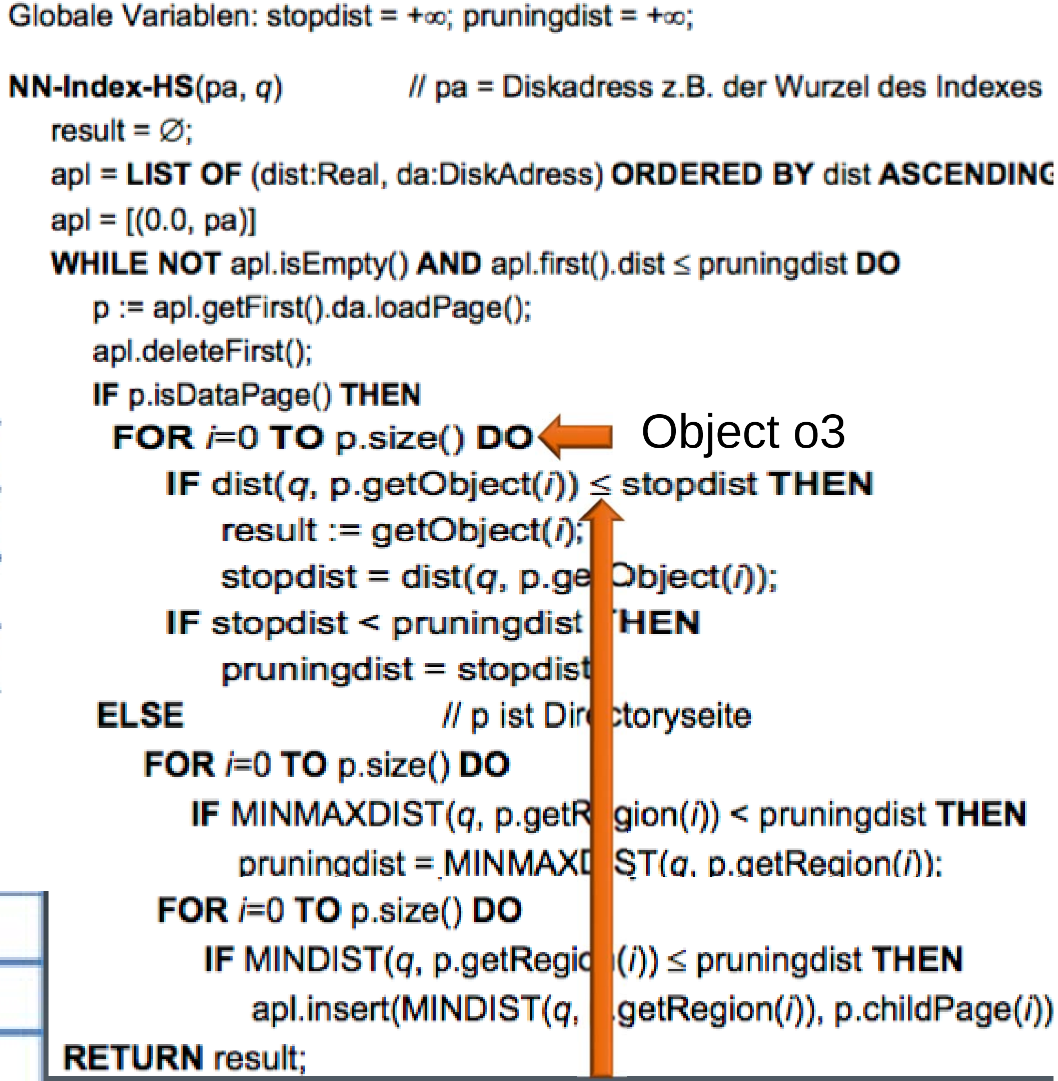
dist(q,o2)=6 <= 5 → Nein, verwirf o2

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22



stopdist = 5 pruningdist= 5



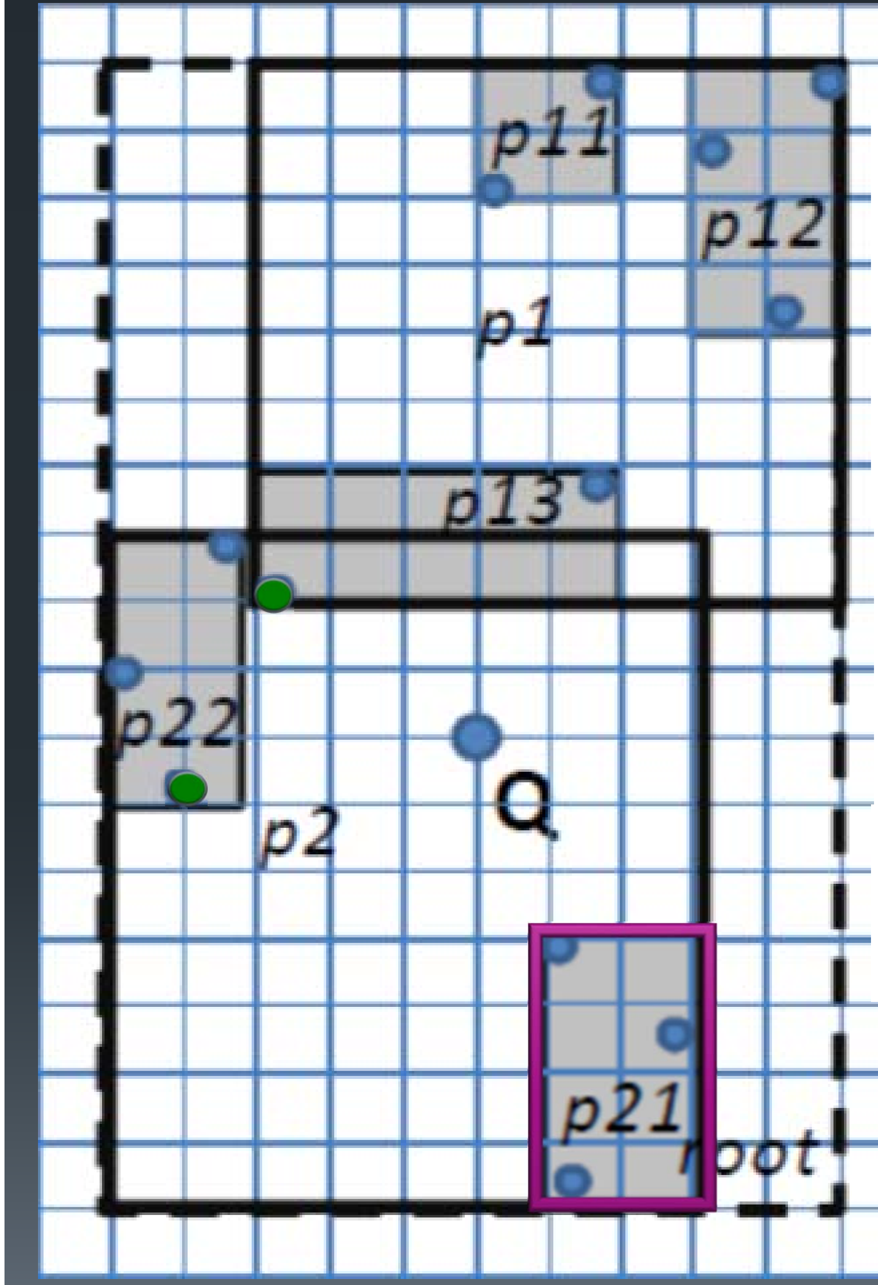
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
       IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
       IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                         // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o3

dist(q,o3)=6.5 <= 5 → Nein, verwirf o3

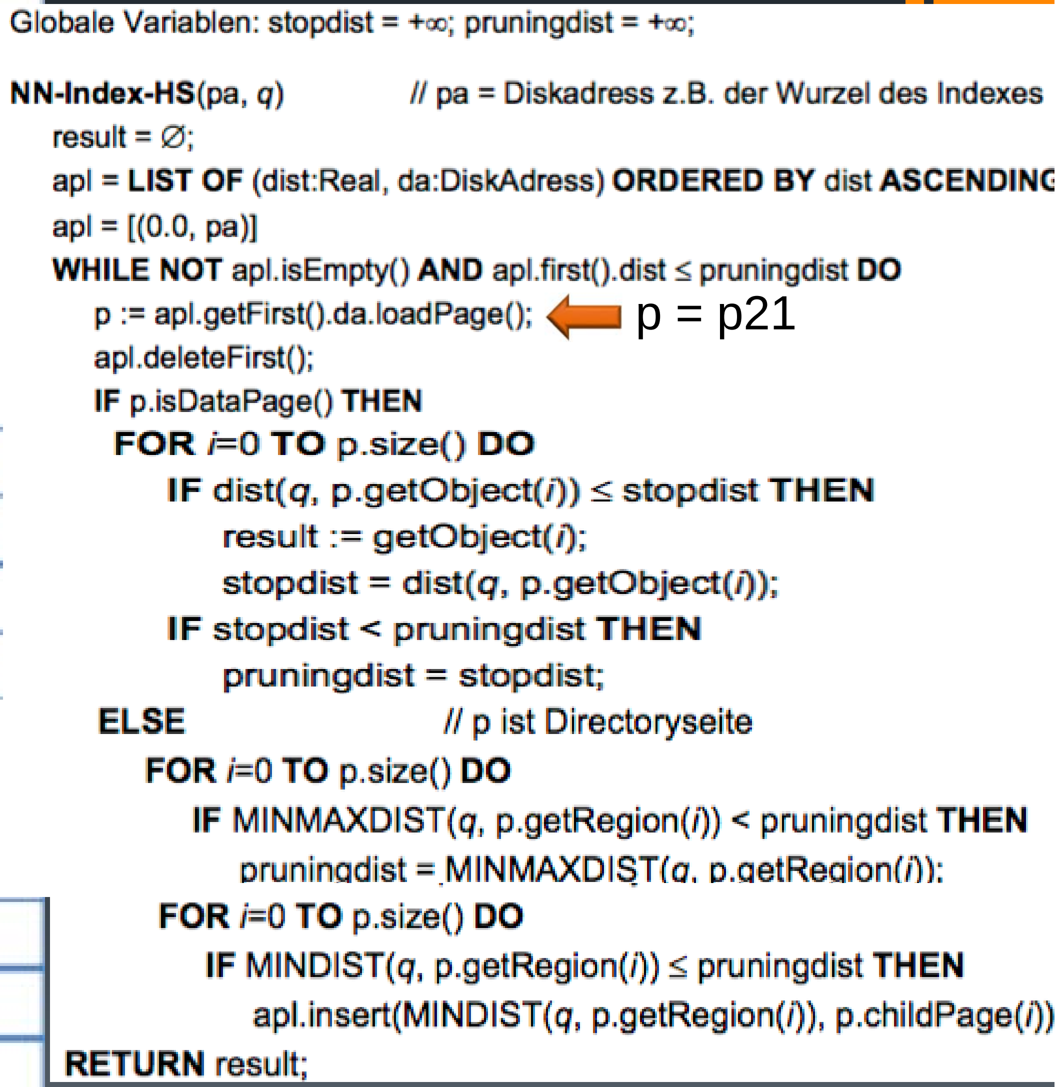
APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 5

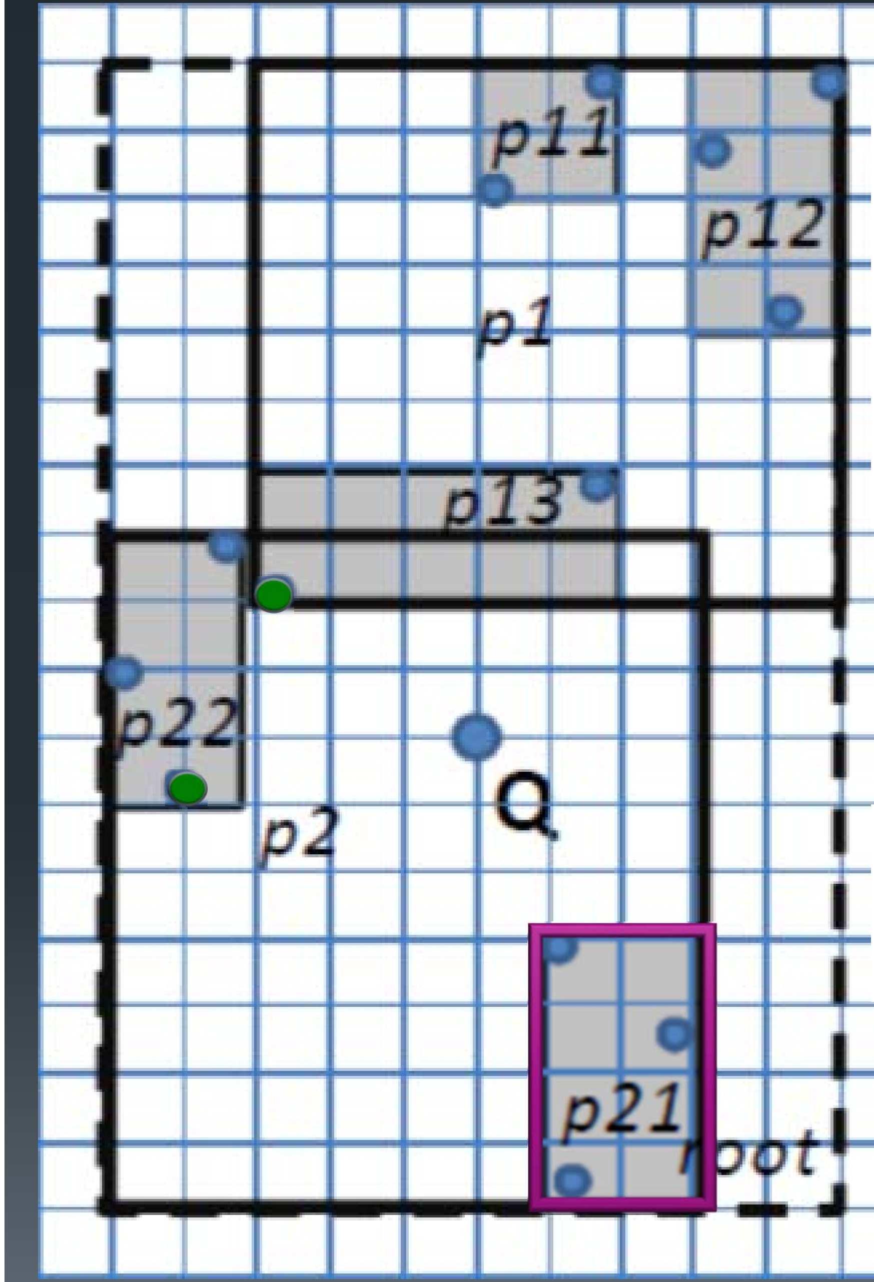
pruningdist= 5



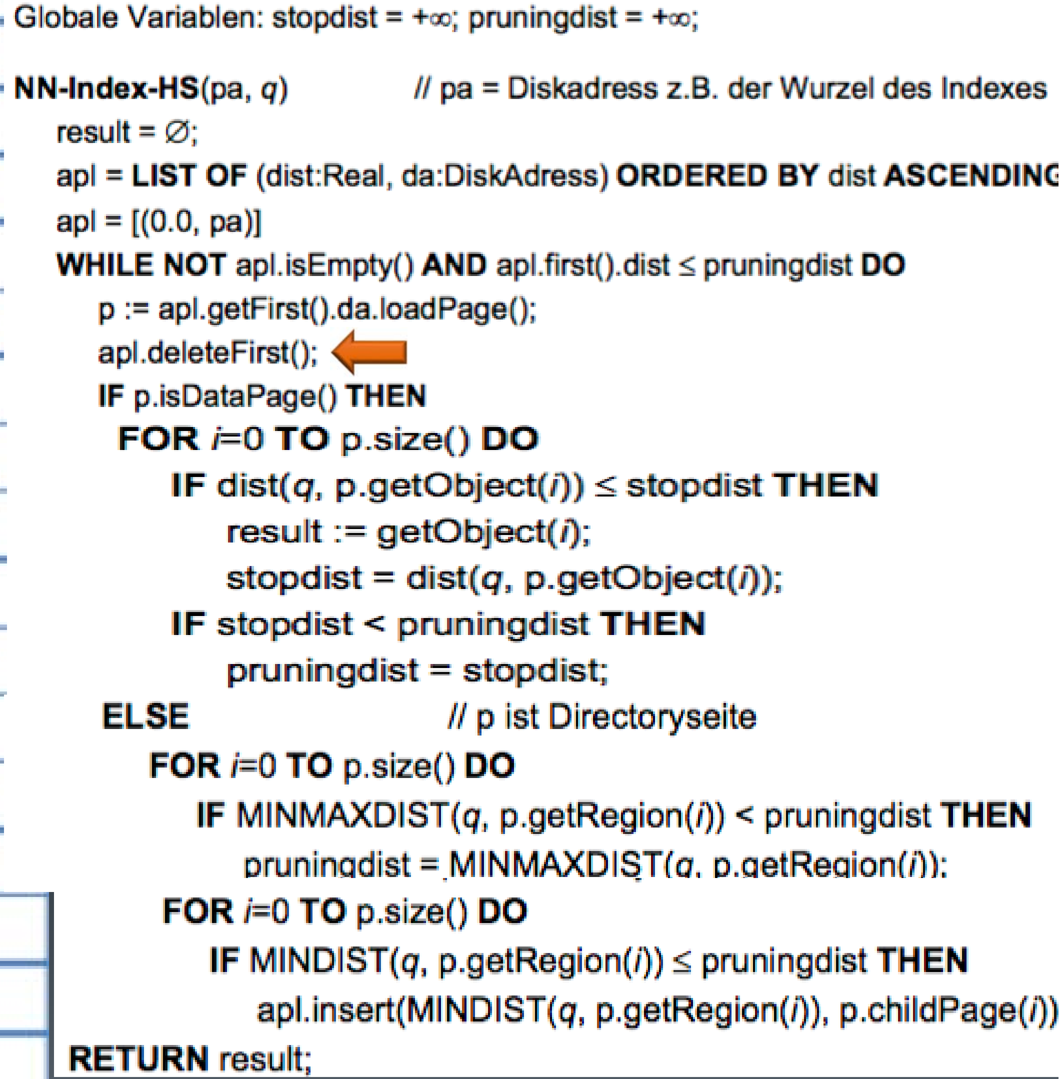
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();   ⇦ p = p21
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

APL= [(4,p21)]

Besuchte Seiten: p2,p1,p13,p22,p21



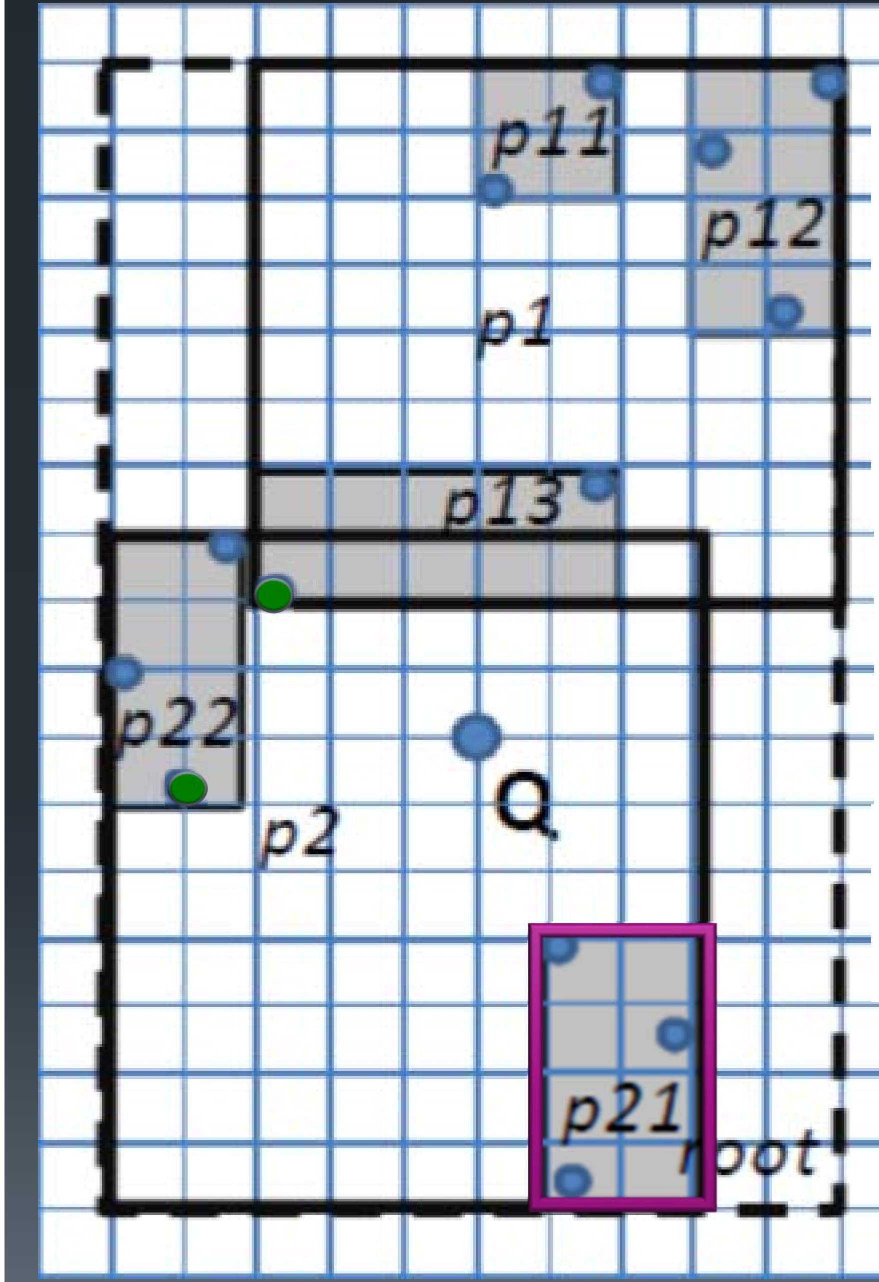
stopdist = 5 pruningdist= 5



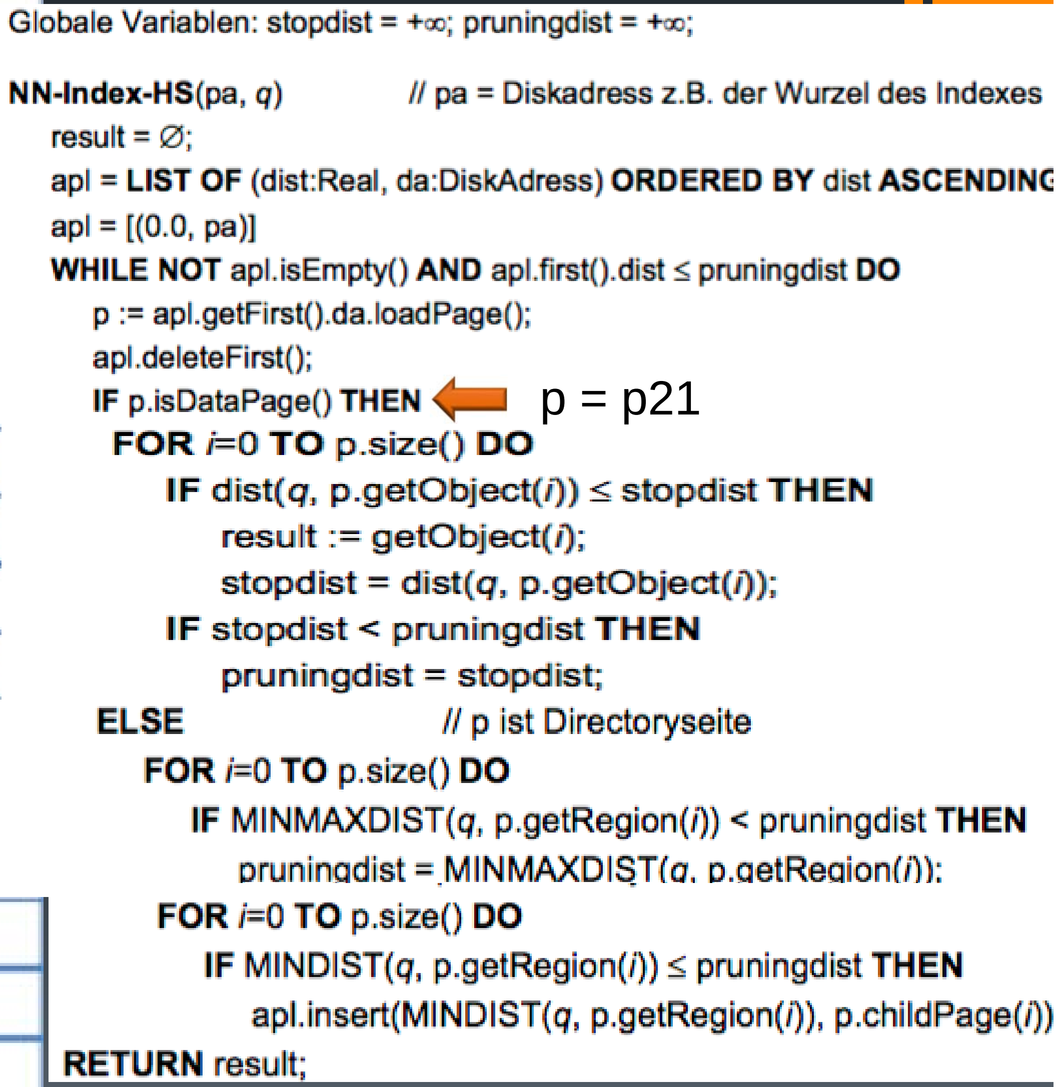
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 5   pruningdist= 5



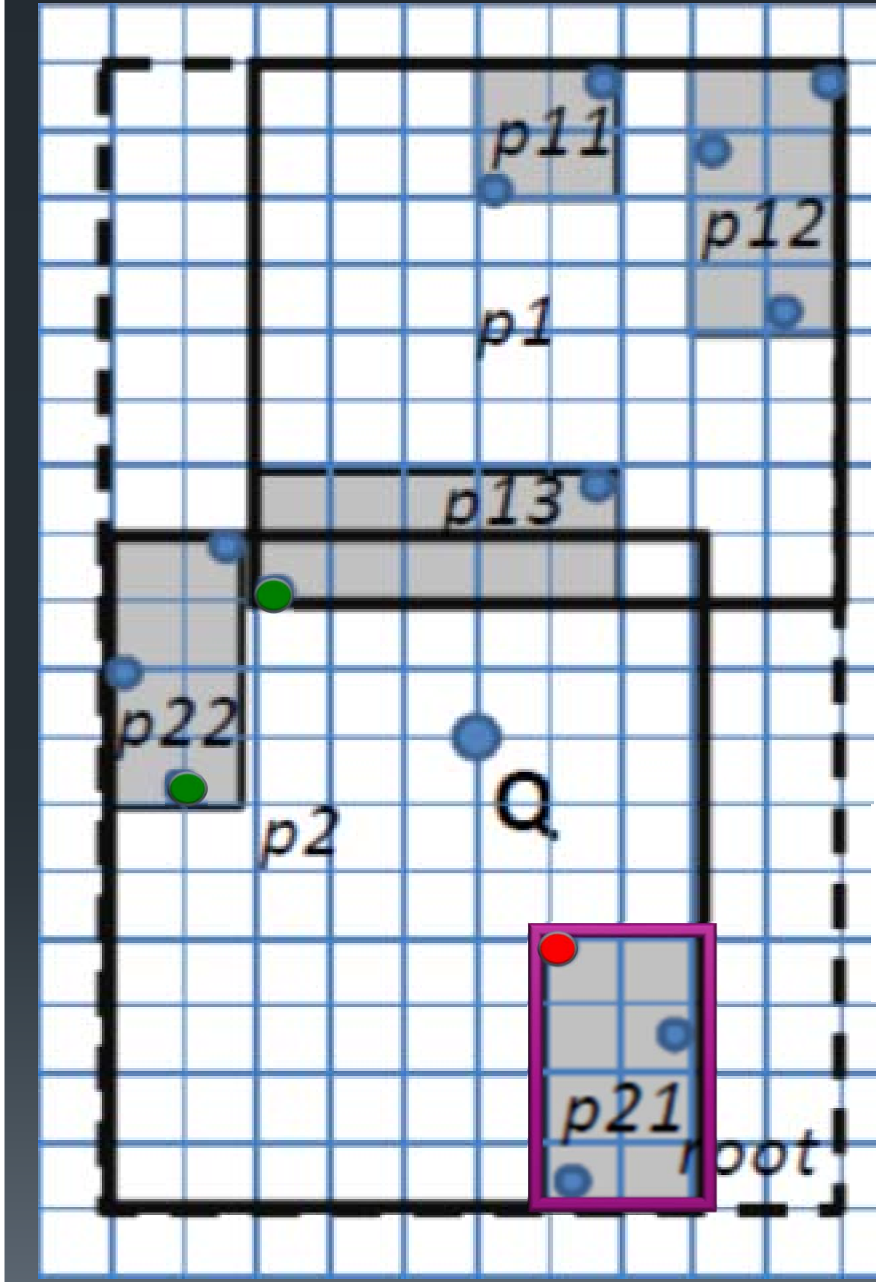
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)          // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDINC
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

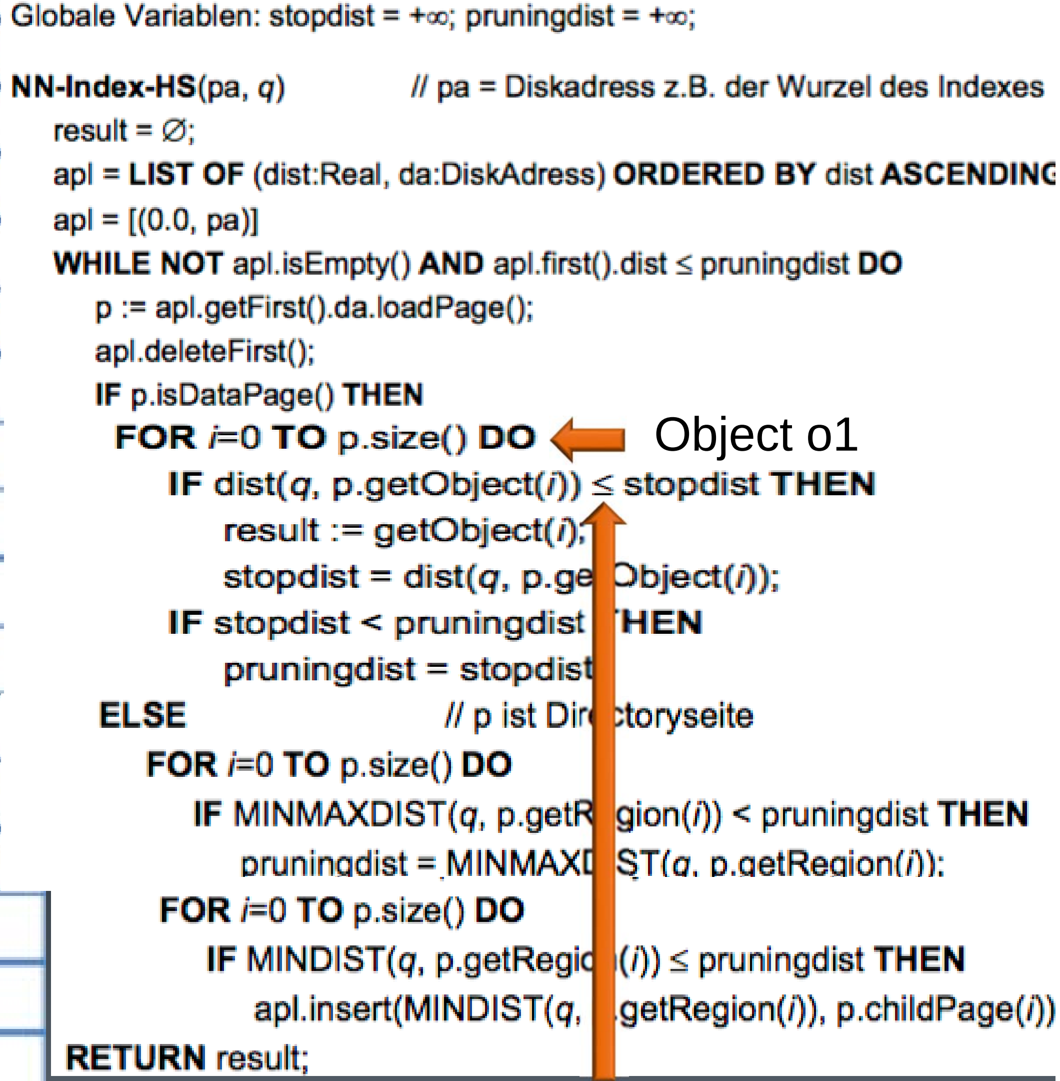
p = p21

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21

stopdist = 5 pruningdist= 5





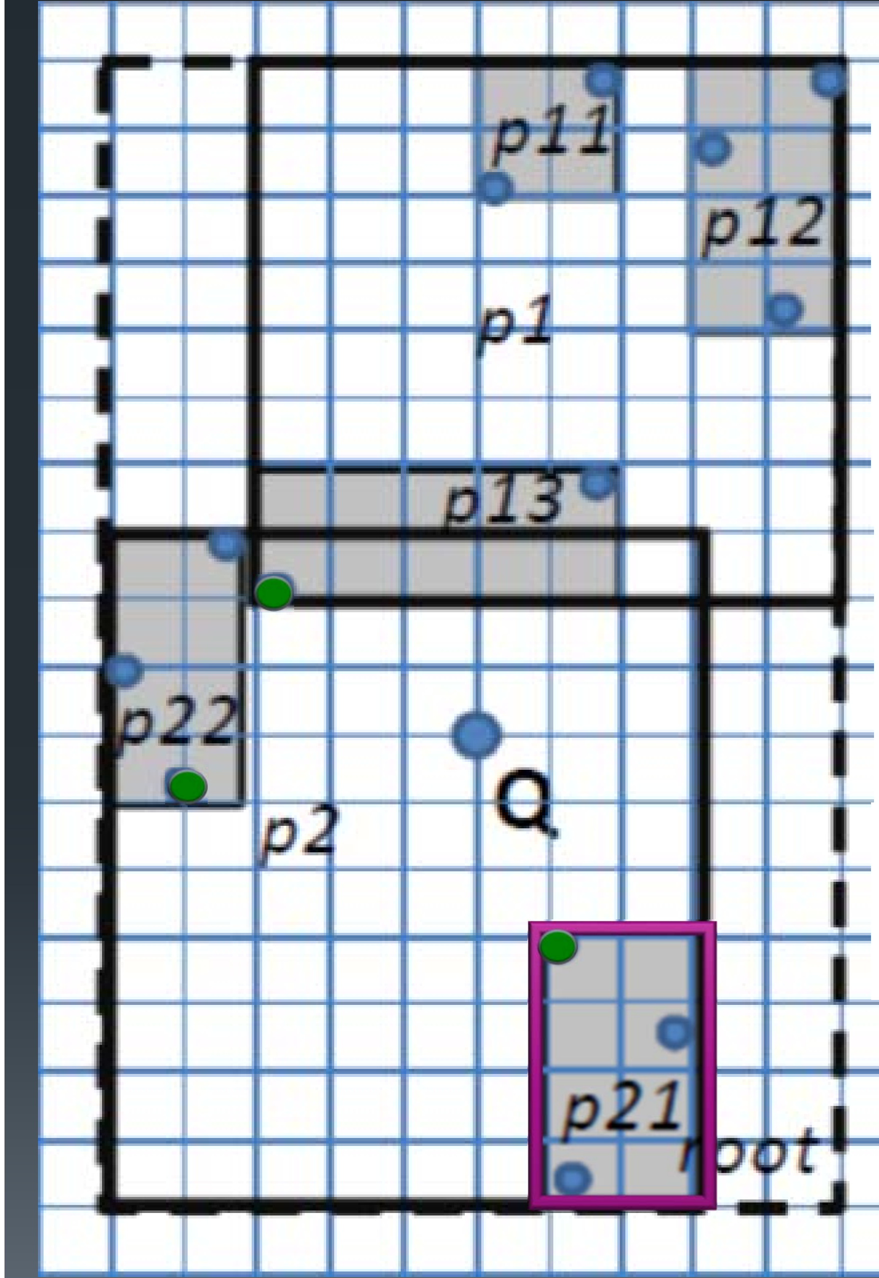
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDIN
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

Object o1

dist(q,o1)=4 <= 5

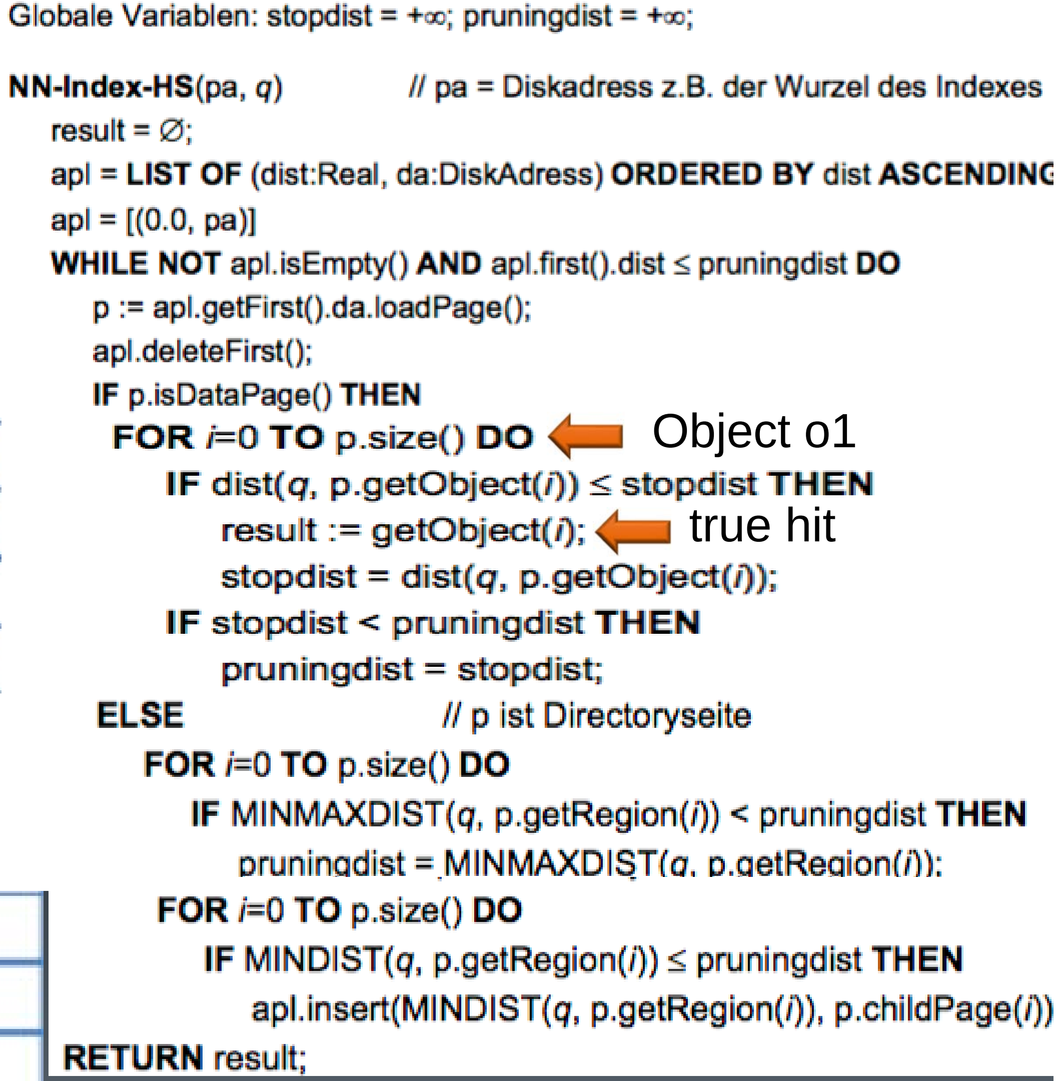
APL= []

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 5

pruningdist= 5



```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

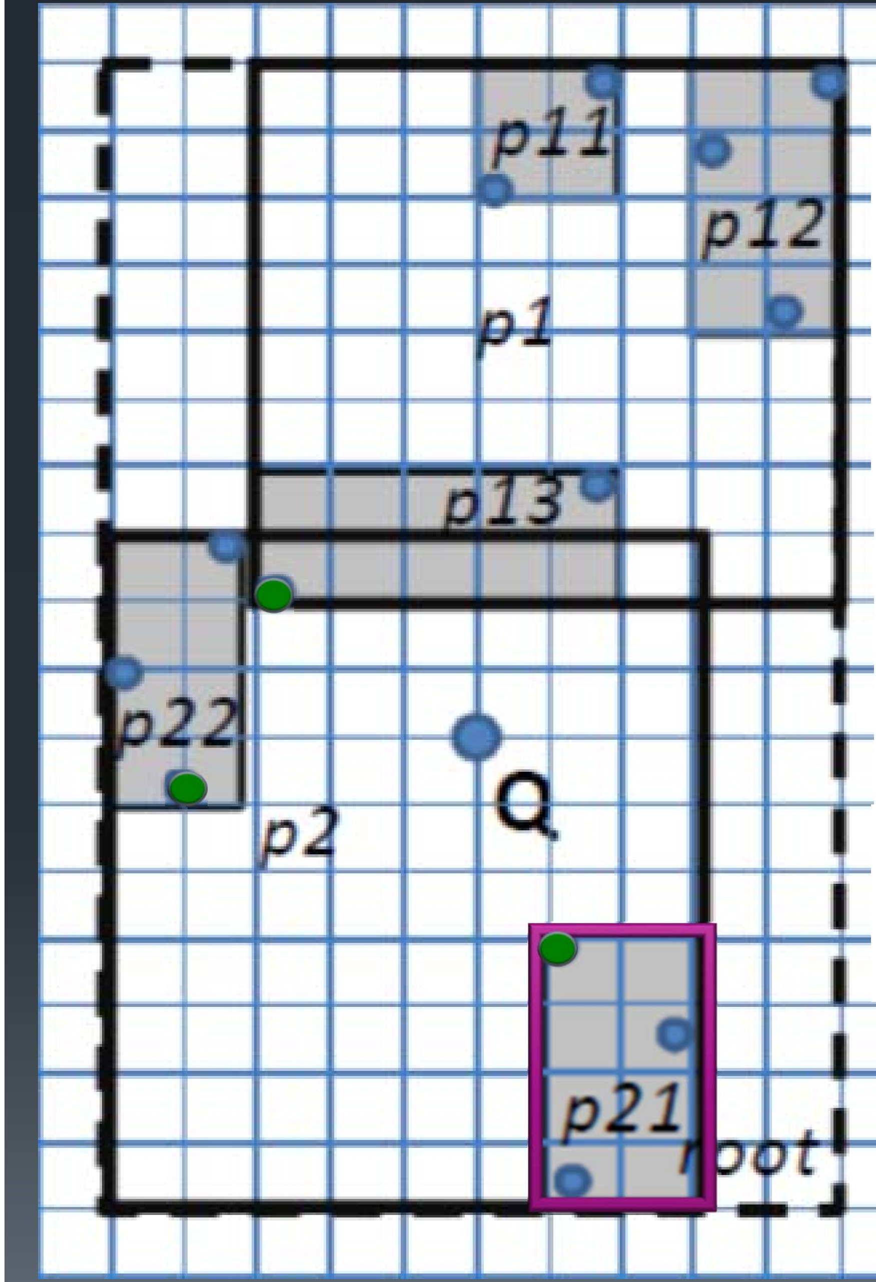
Object o1

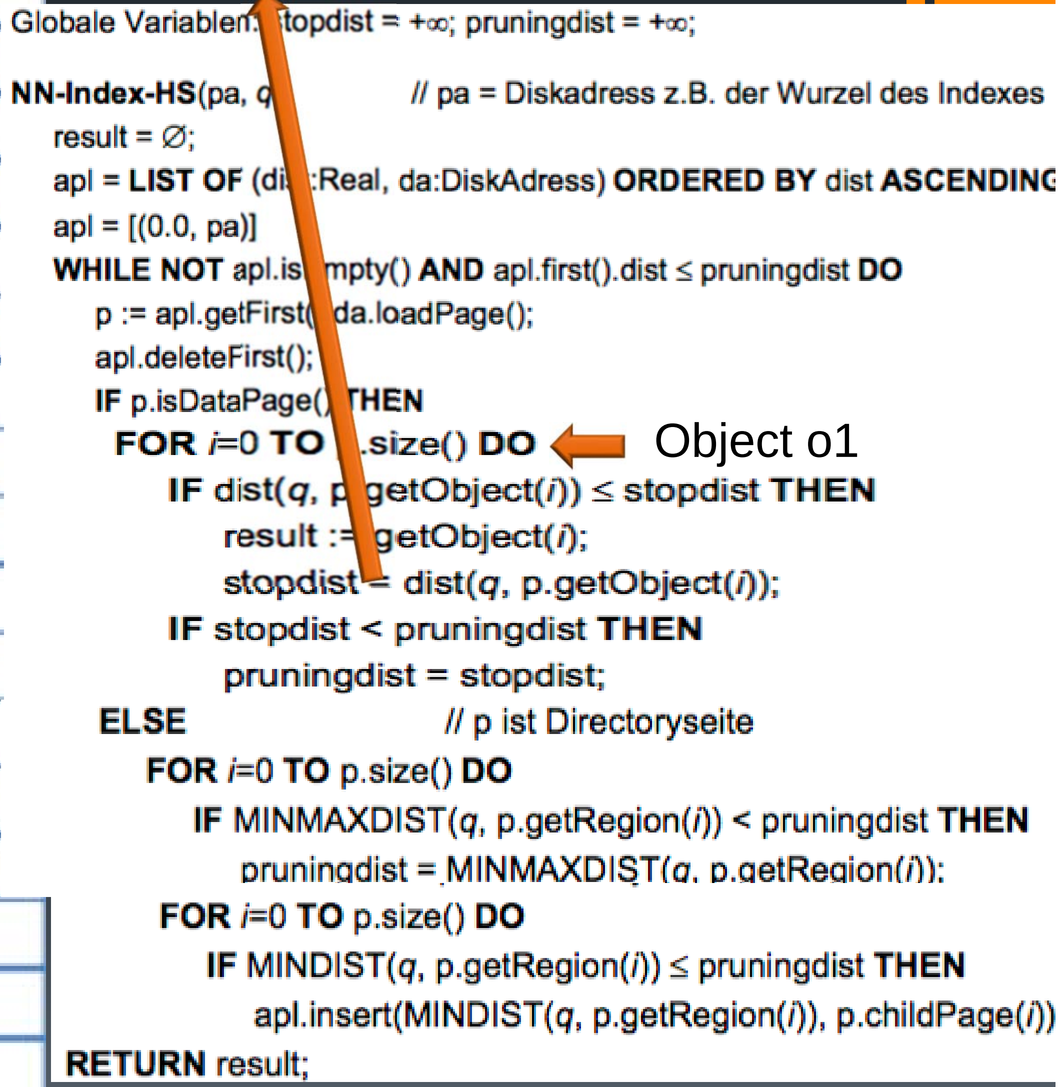
true hit

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21

stopdist = 4 pruningdist= 5



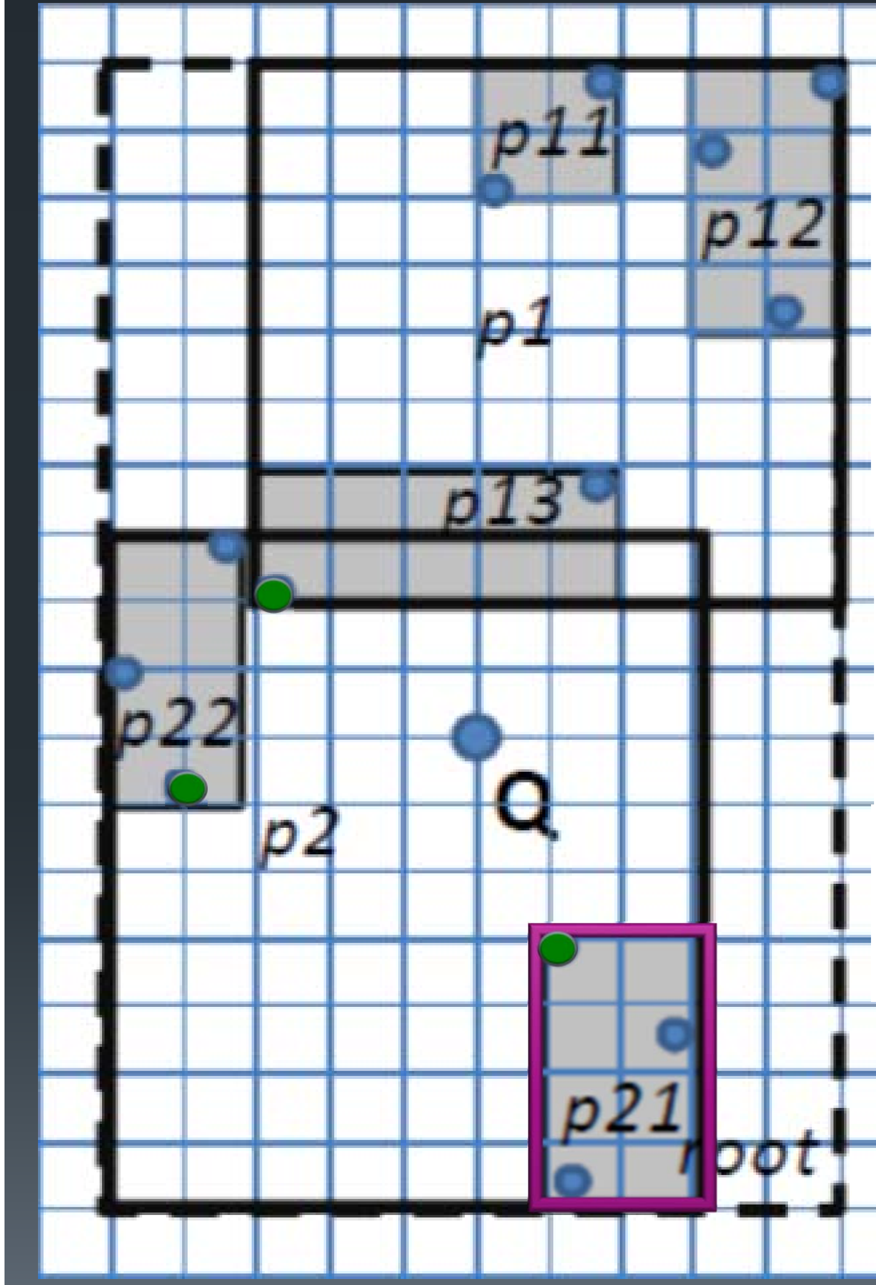


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := p.getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                       // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

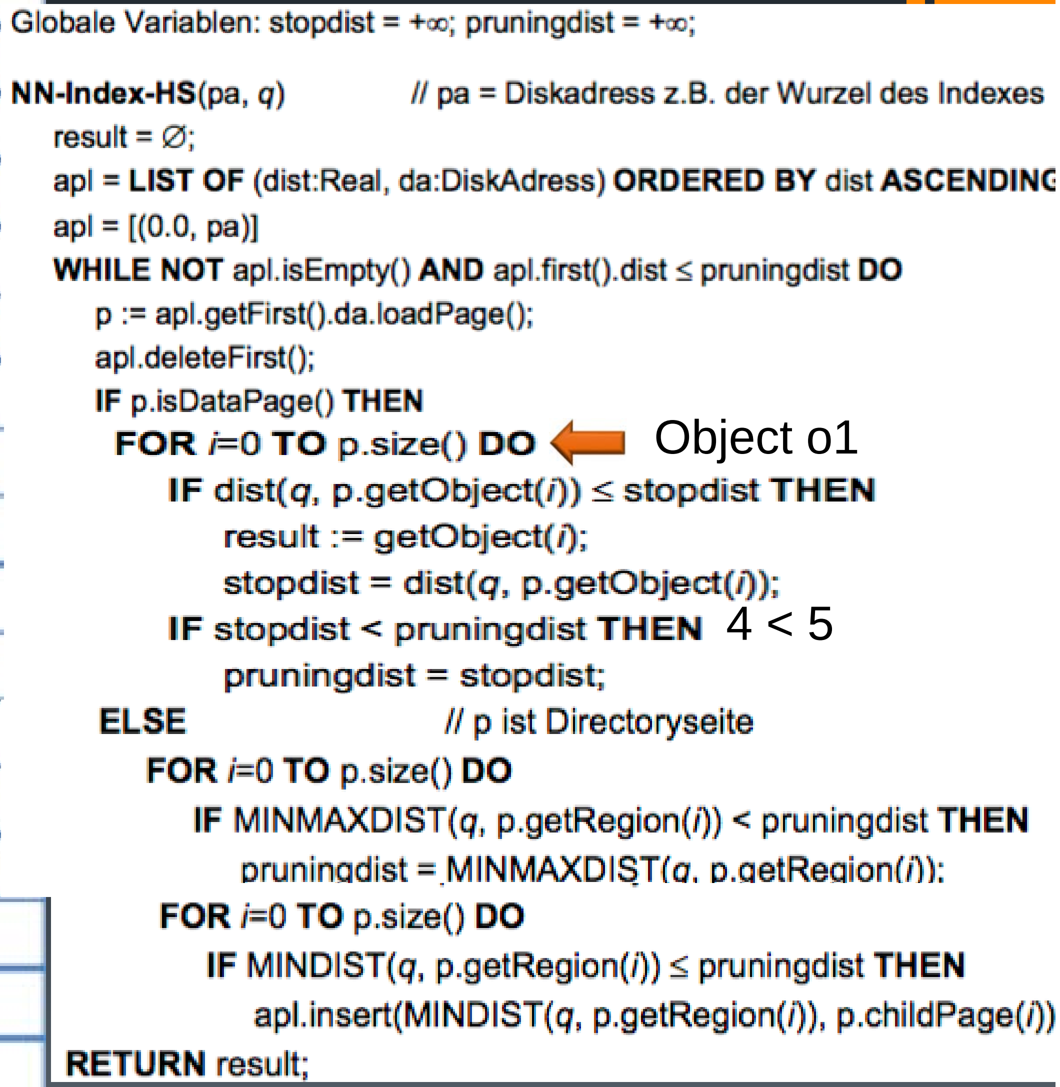
Object o1

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 4 pruningdist= 5

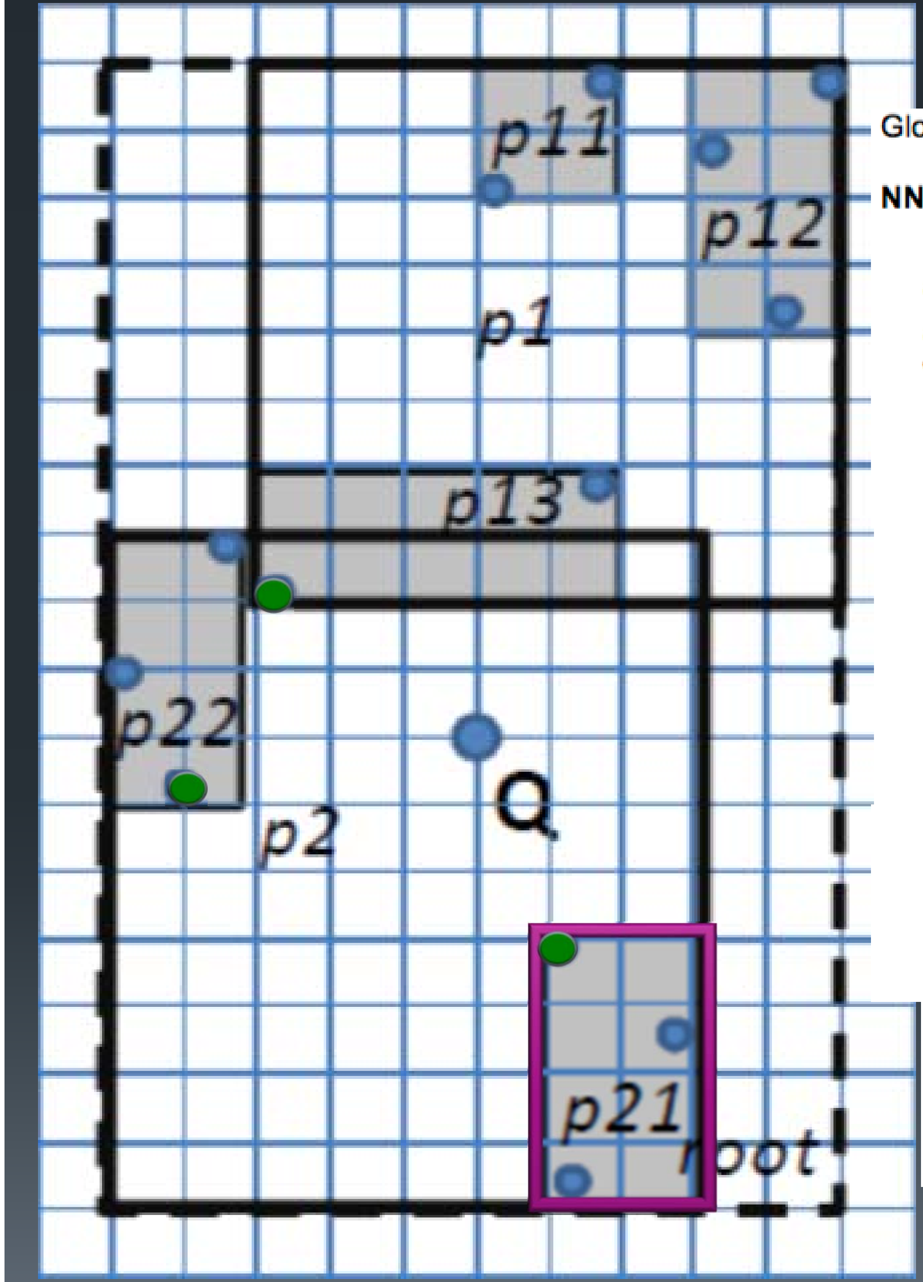


```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO   ⇦  Object o1
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN   4 < 5
          pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21

stopdist = 4 pruningdist= 4



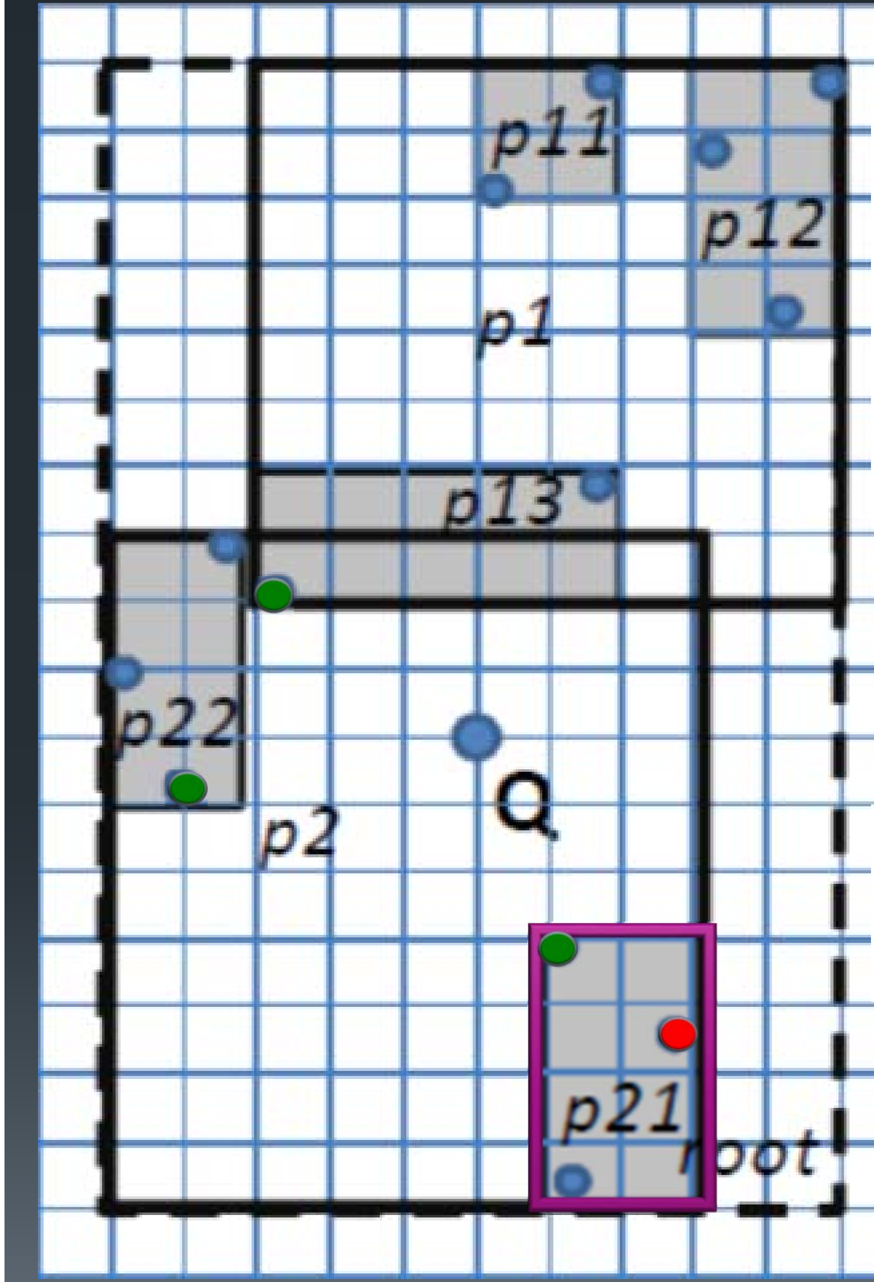
```
Globale Variablen: stopdist = +∞; pruningdist = +∞

NN-Index-HS(pa, q)            // pa = Diskadresse z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
     FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
           result := getObject(i);
           stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
           pruningdist = stopdist;
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
 RETURN result;
```

Object o1

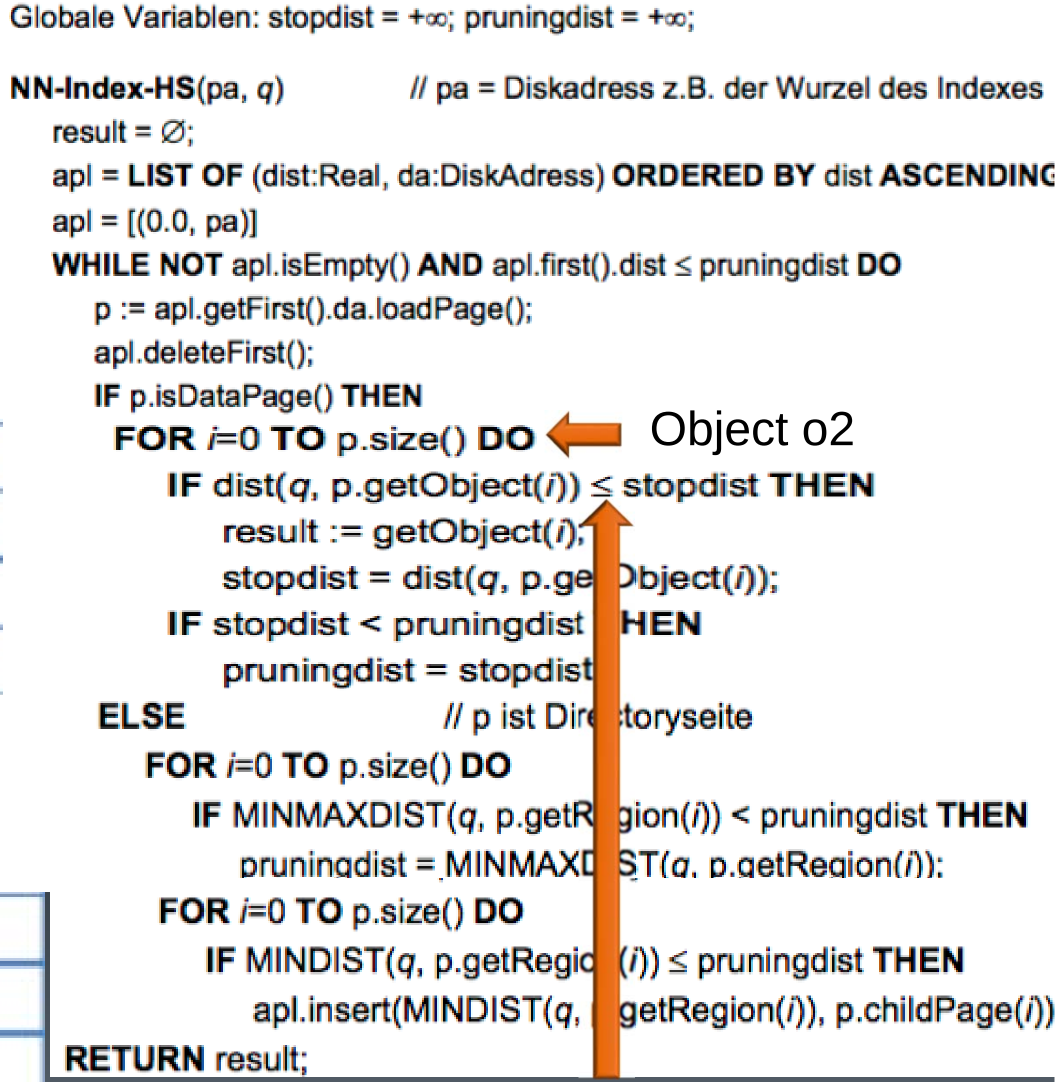
4 < 5

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 4 pruningdist= 4



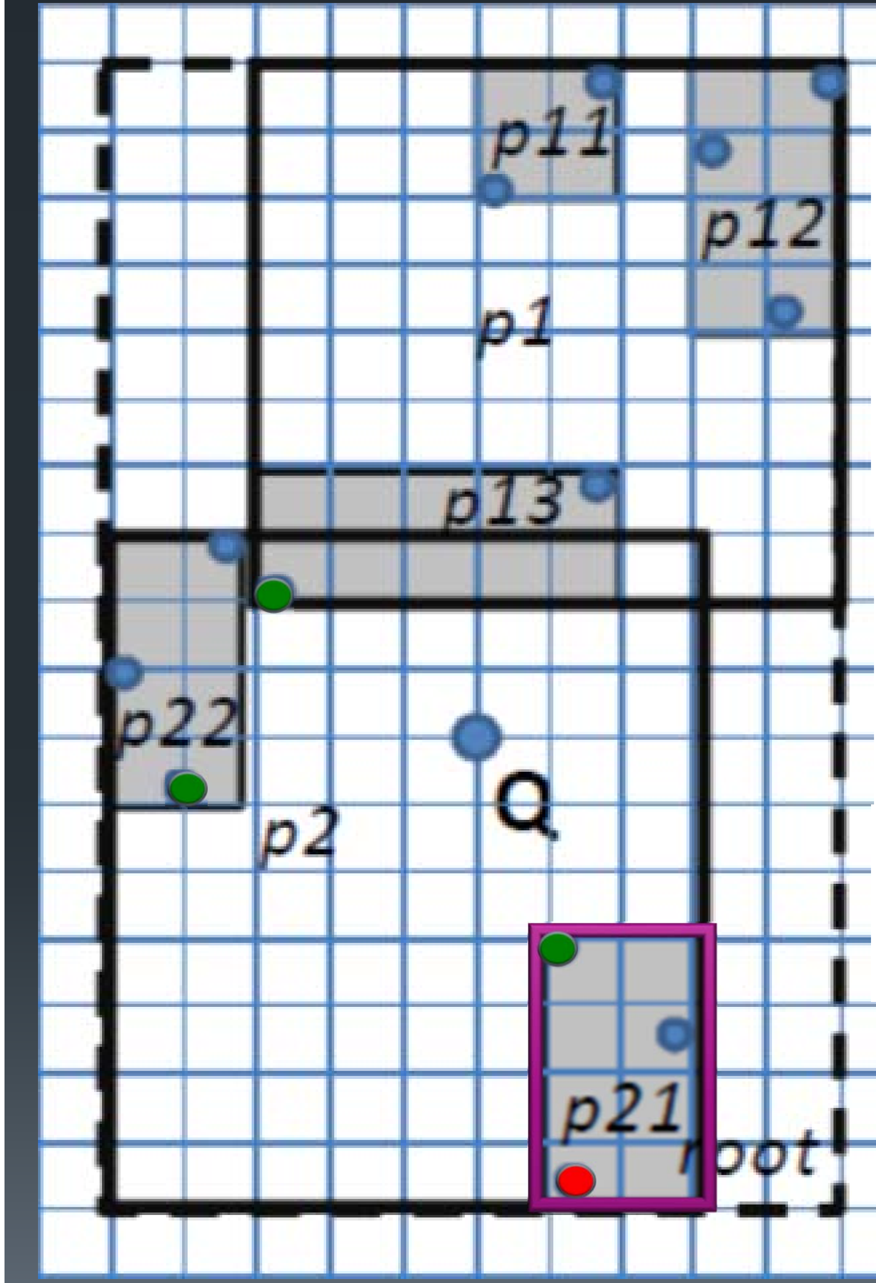
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
RETURN result;
```

Object o2

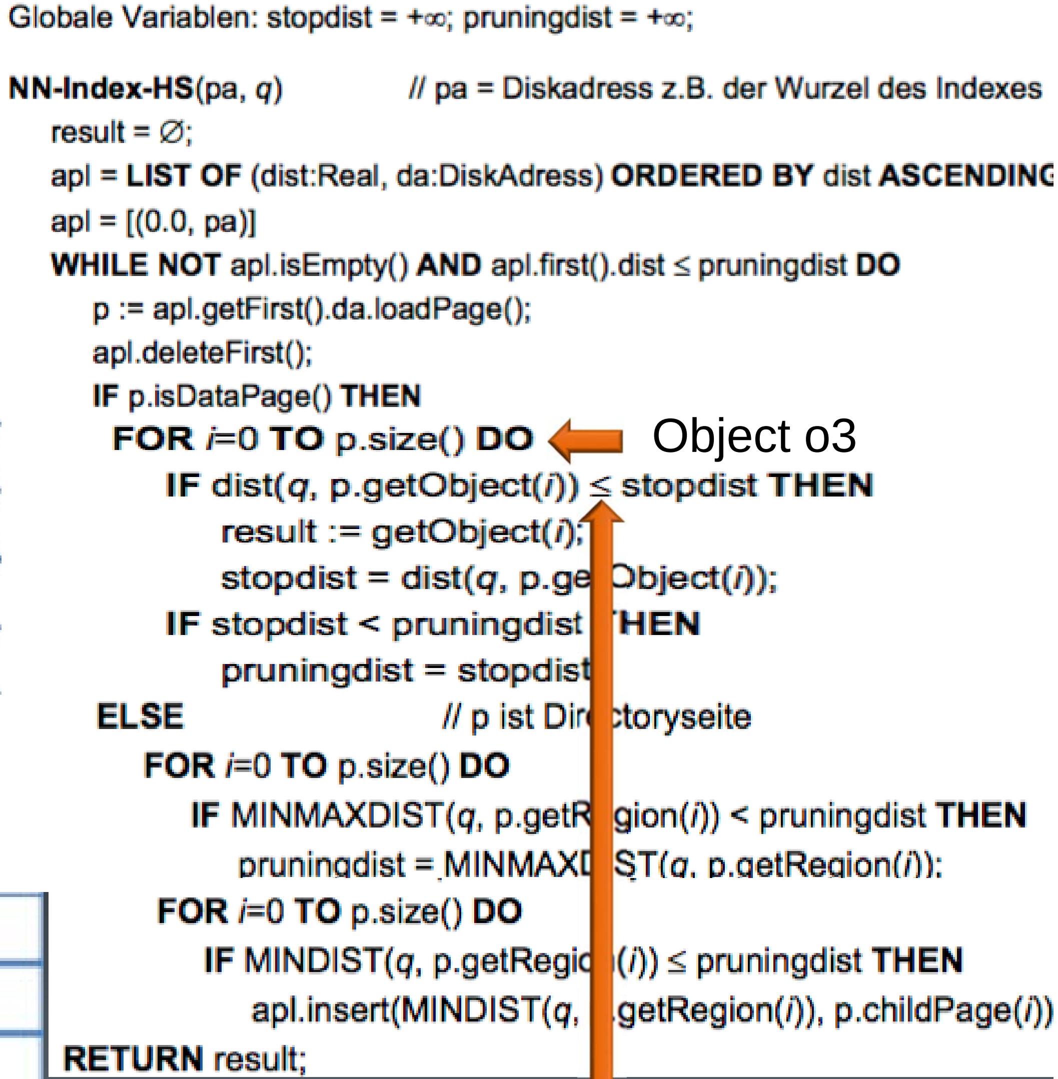
dist(q,o2)=7.5 <= 4 → Nein, verwirf o2

APL= []

Besuchte Seiten: p2,p1,p13,p22,p21



stopdist = 4 pruningdist= 4



```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)           // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist
    ELSE                        // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

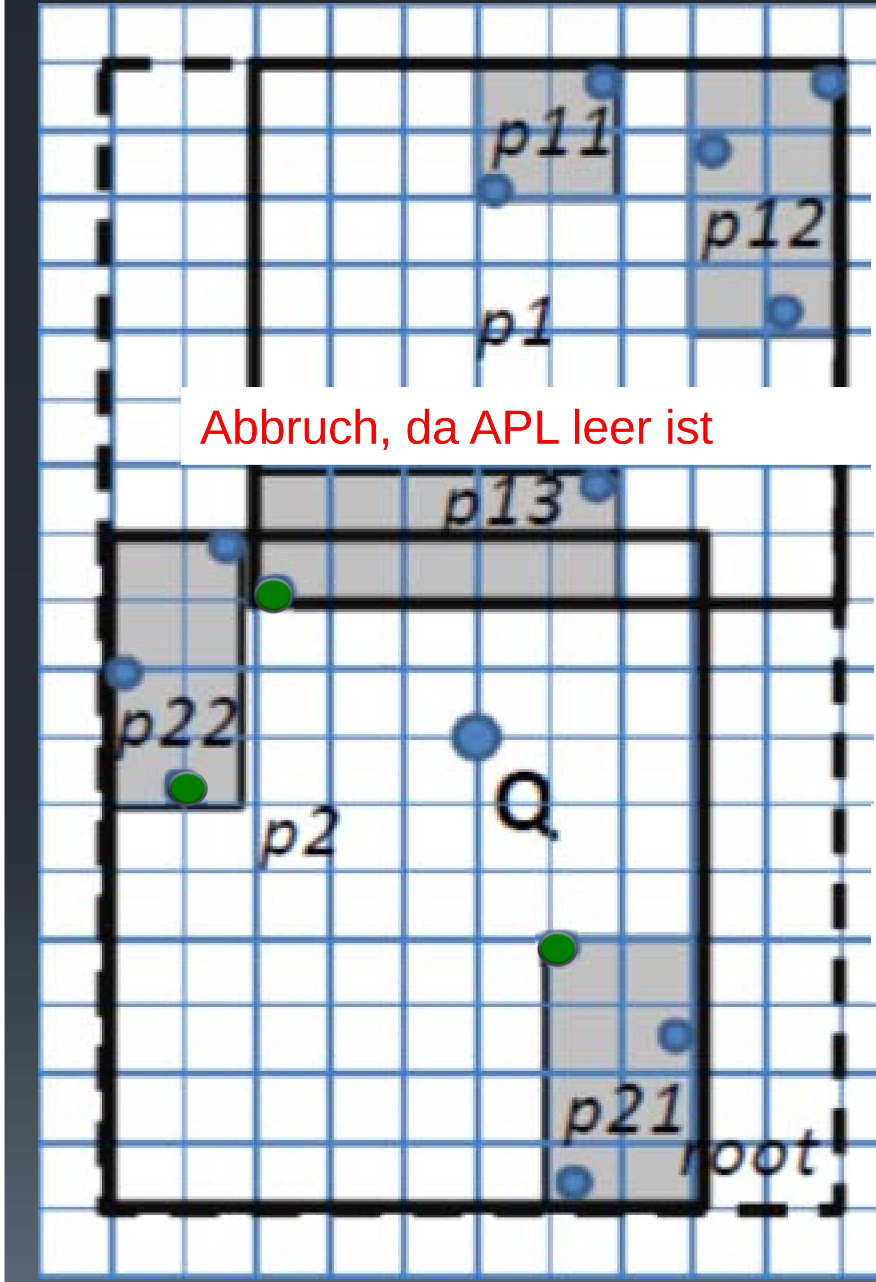
Object o3

dist(q,o3)=8.5 <= 4 → Nein, verwirf o3

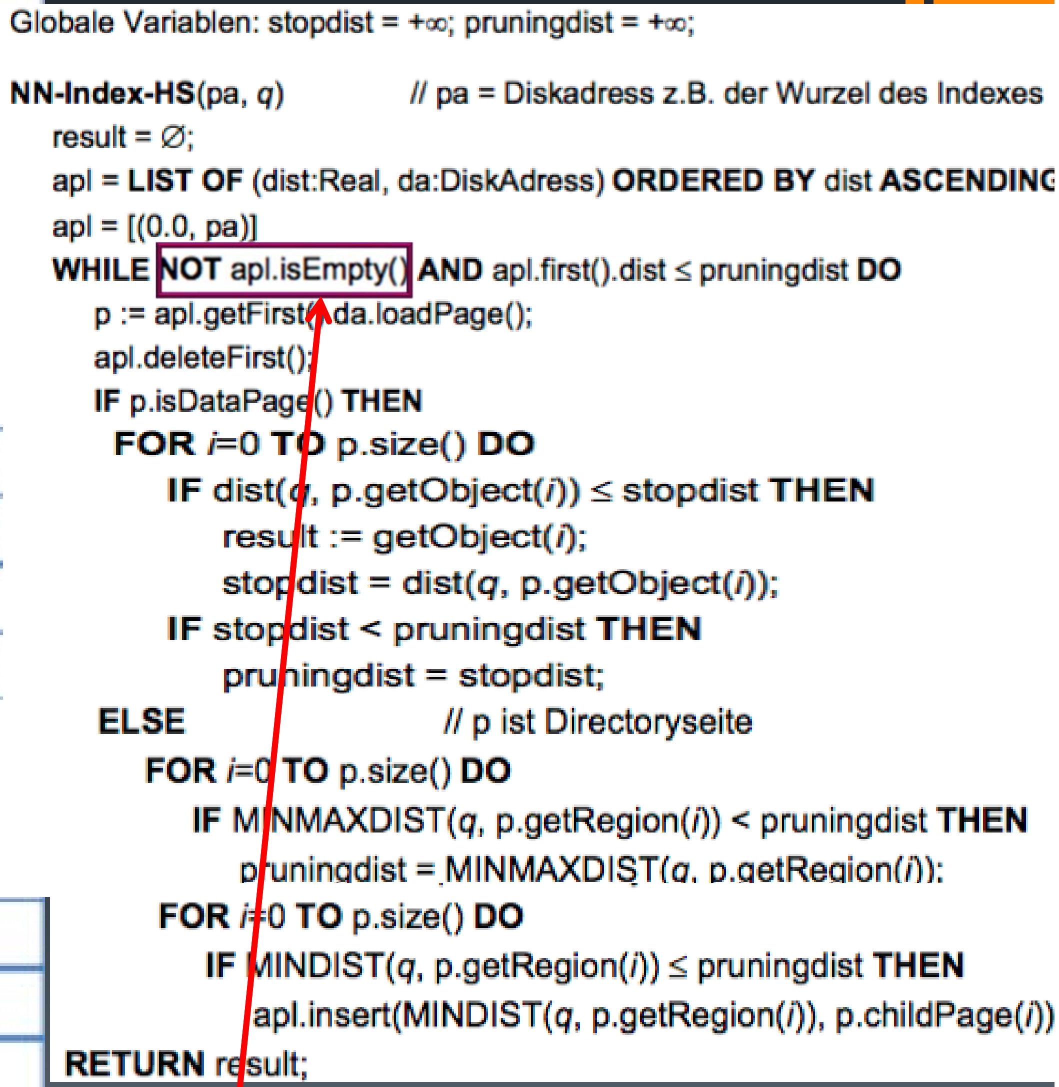
APL= []

Besuchte Seiten: p2,p1,p13,p22,p21

stopdist = 4 pruningdist= 4



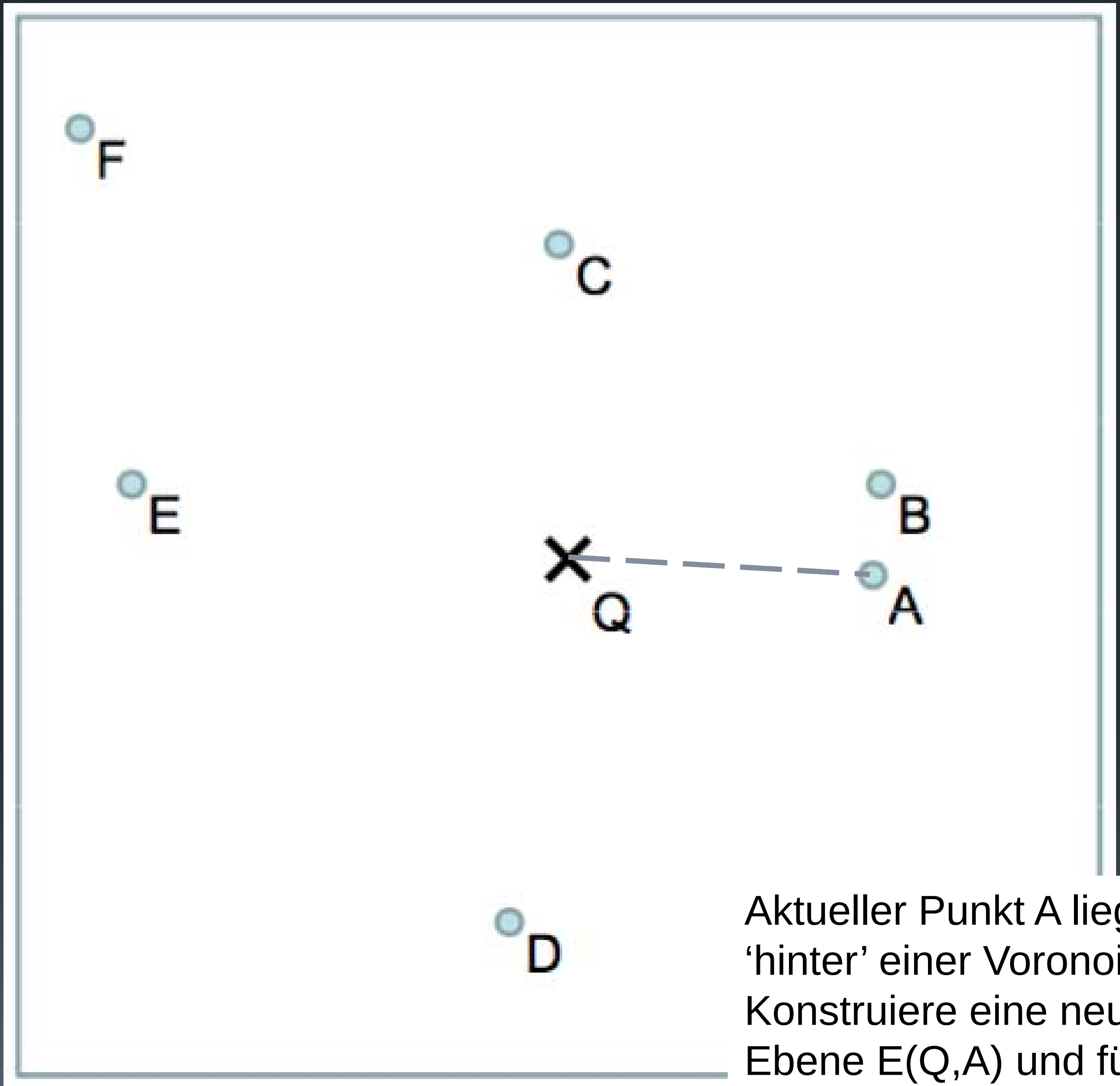
Abbruch, da APL leer ist



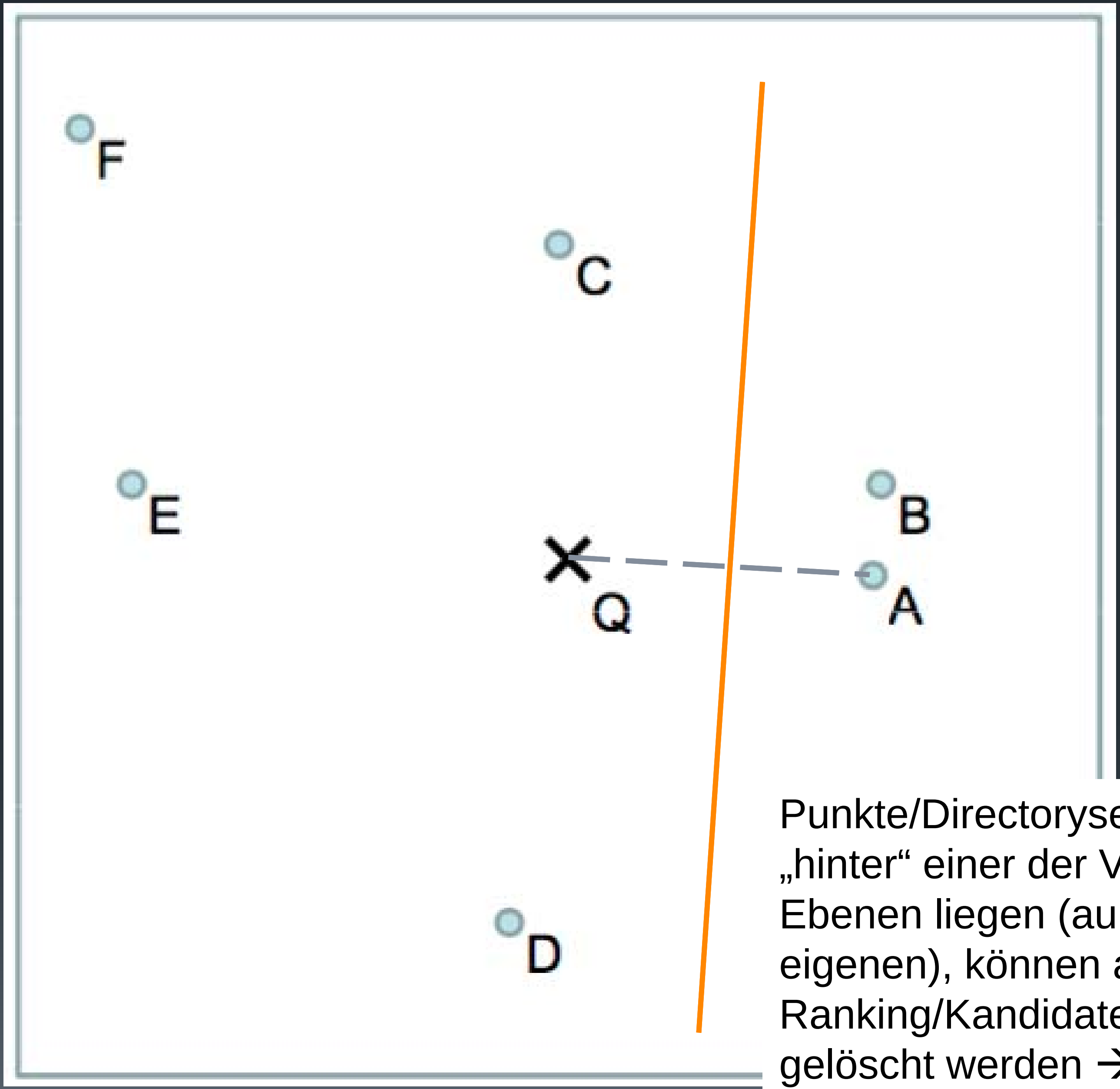
```
Globale Variablen: stopdist = +∞; pruningdist = +∞;

NN-Index-HS(pa, q)            // pa = Diskadress z.B. der Wurzel des Indexes
  result = ∅;
  apl = LIST OF (dist:Real, da:DiskAdress) ORDERED BY dist ASCENDING
  apl = [(0.0, pa)]
  WHILE NOT apl.isEmpty() AND apl.first().dist ≤ pruningdist DO
    p := apl.getFirst().da.loadPage();
    apl.deleteFirst();
    IF p.isDataPage() THEN
      FOR i=0 TO p.size() DO
        IF dist(q, p.getObject(i)) ≤ stopdist THEN
          result := getObject(i);
          stopdist = dist(q, p.getObject(i));
        IF stopdist < pruningdist THEN
          pruningdist = stopdist;
    ELSE                          // p ist Directoryseite
      FOR i=0 TO p.size() DO
        IF MINMAXDIST(q, p.getRegion(i)) < pruningdist THEN
          pruningdist = MINMAXDIST(q, p.getRegion(i));
      FOR i=0 TO p.size() DO
        IF MINDIST(q, p.getRegion(i)) ≤ pruningdist THEN
          apl.insert(MINDIST(q, p.getRegion(i)), p.childPage(i))
  RETURN result;
```

APL= []

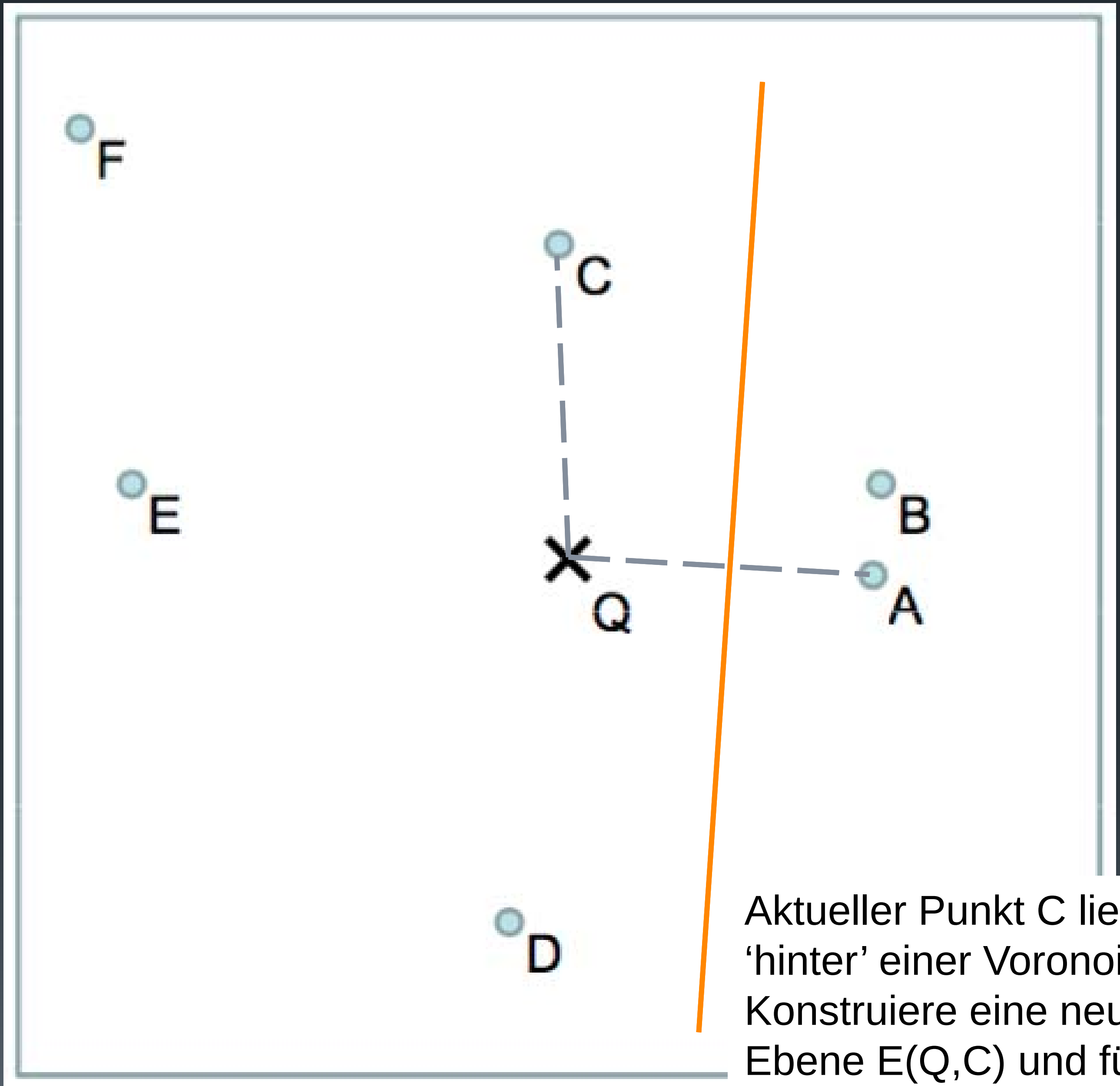
A5-3



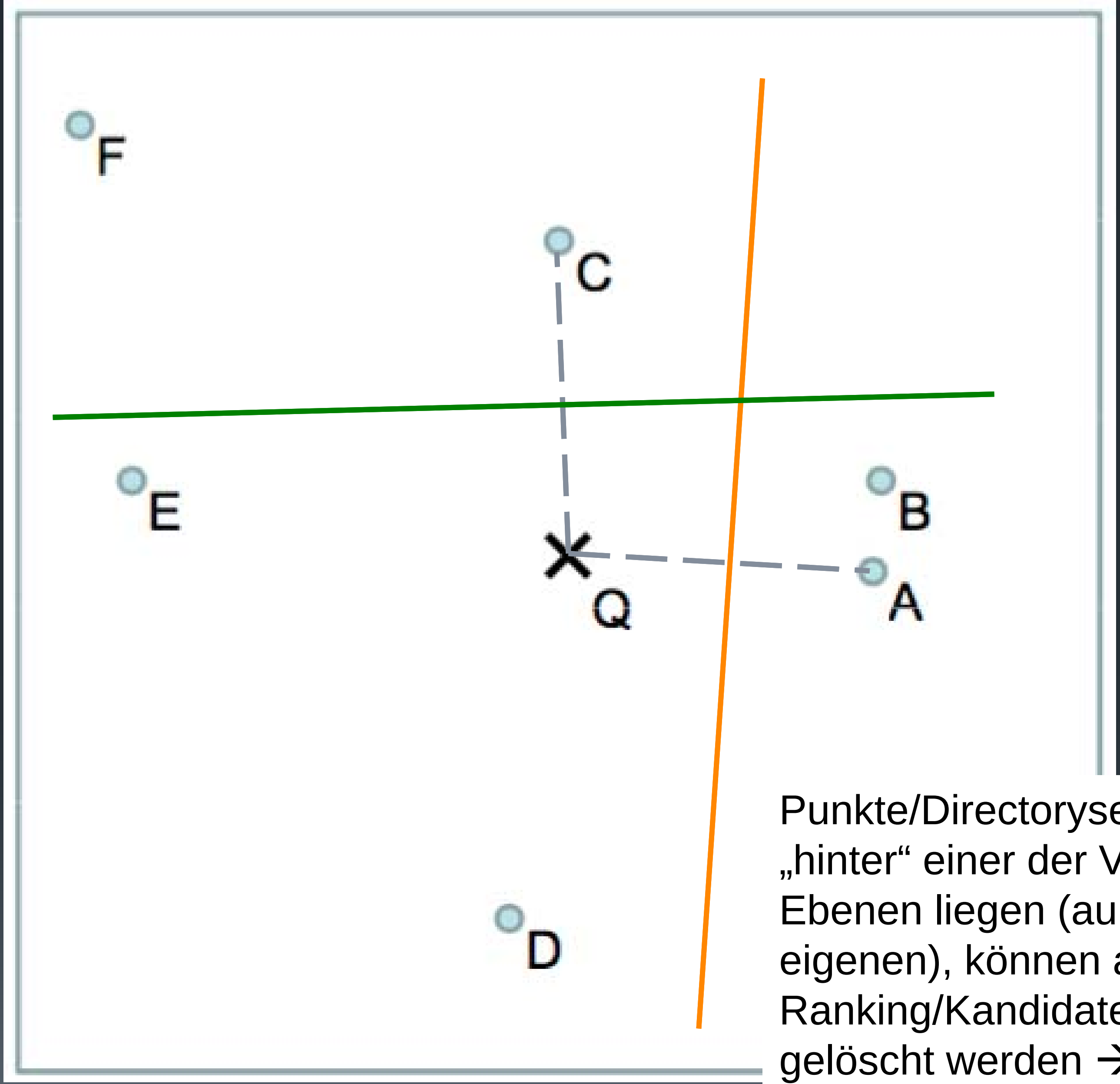
Aktueller Punkt A liegt nicht ‘hinter’ einer Voronoi-Ebene. Konstruiere eine neue Voronoi-Ebene E(Q,A) und füge A zur Kandidatenmenge hinzu



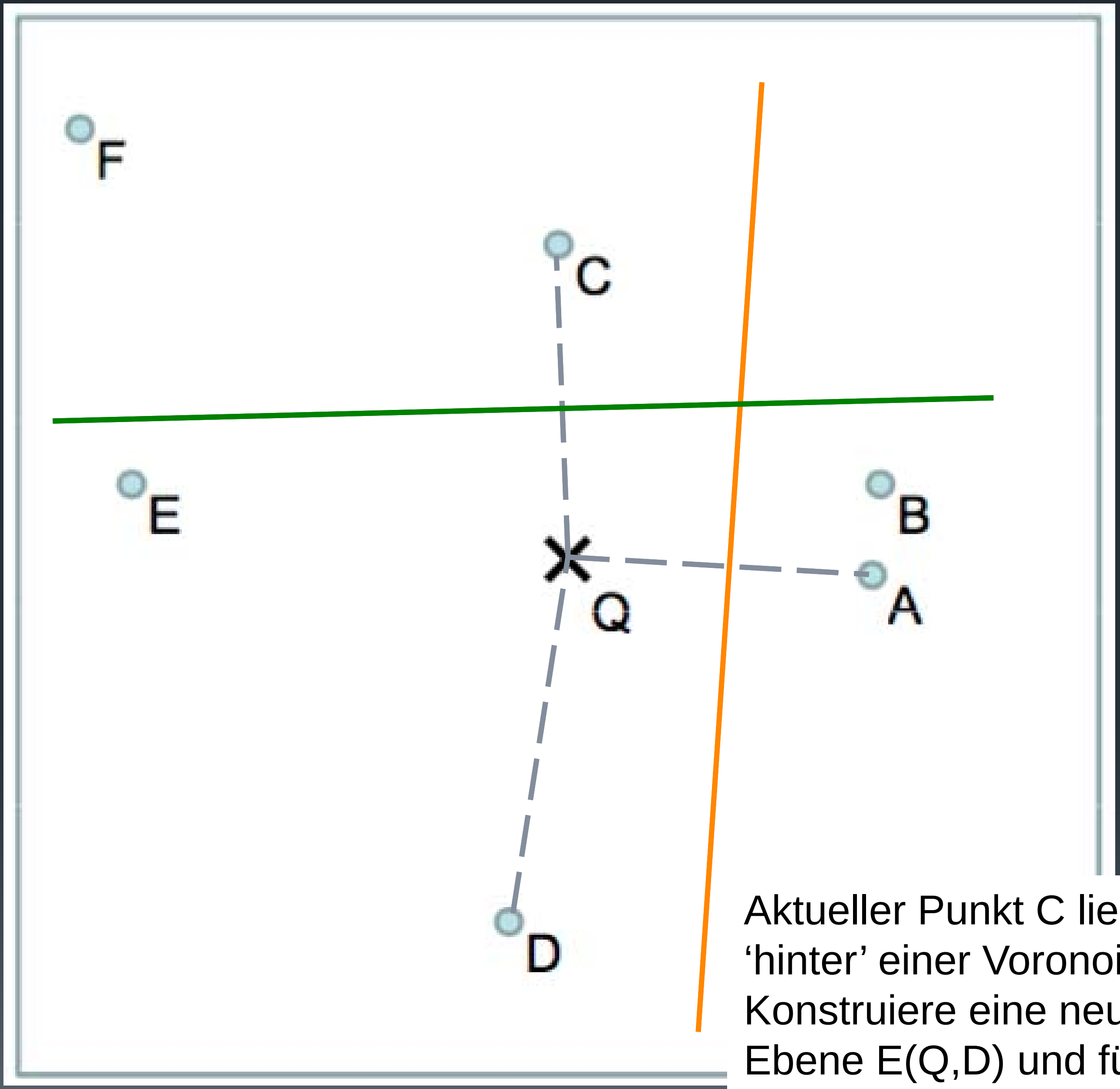
Punkte/Directoryseiten, die „hinter“ einer der Voronoi-Ebenen liegen (außer der eigenen), können aus dem Ranking/Kandidatenmenge gelöscht werden → Entferne Punkt B



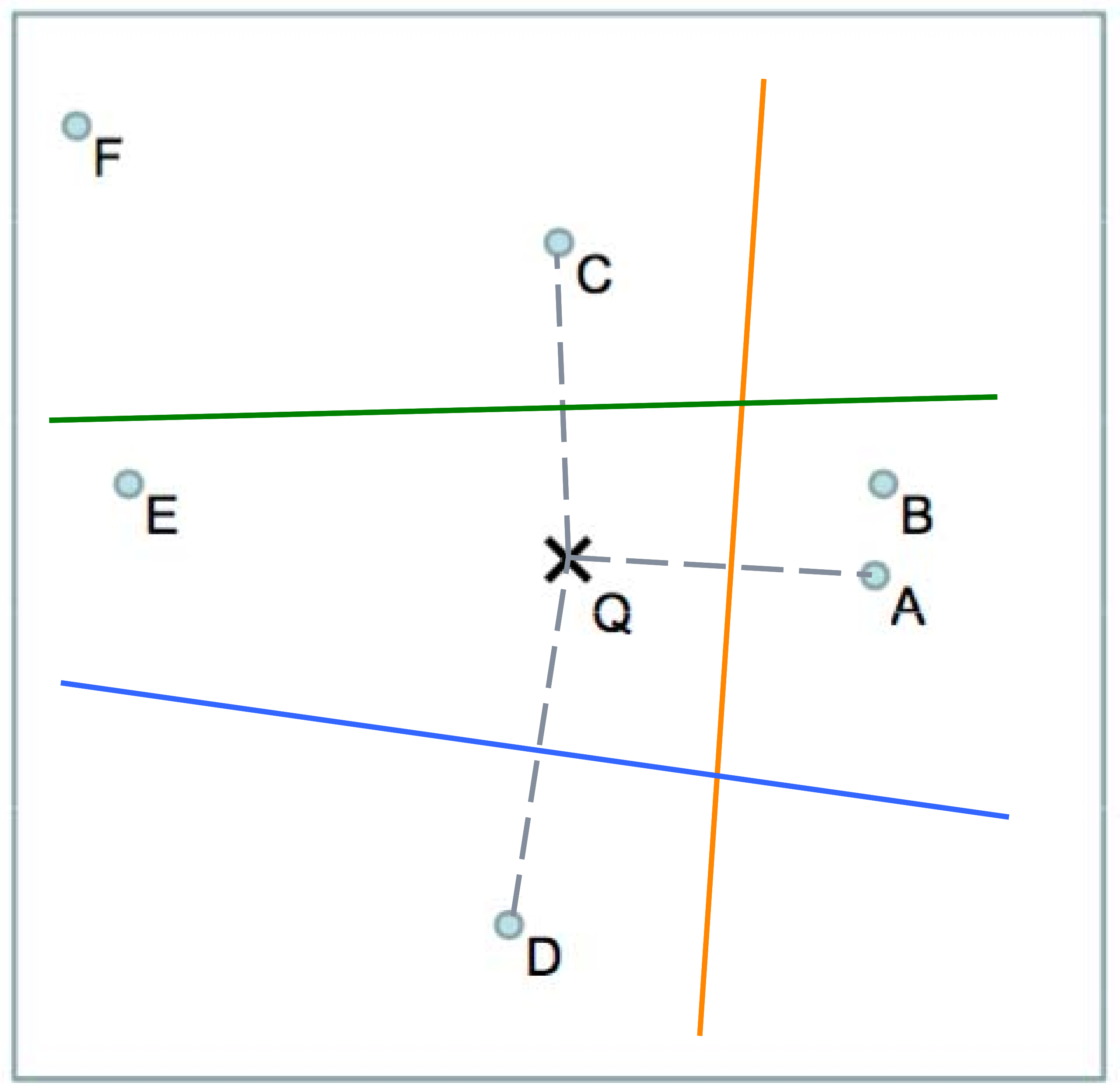
Aktueller Punkt C liegt nicht ‘hinter’ einer Voronoi-Ebene. Konstruiere eine neue Voronoi-Ebene E(Q,C) und füge C zur Kandidatenmenge hinzu

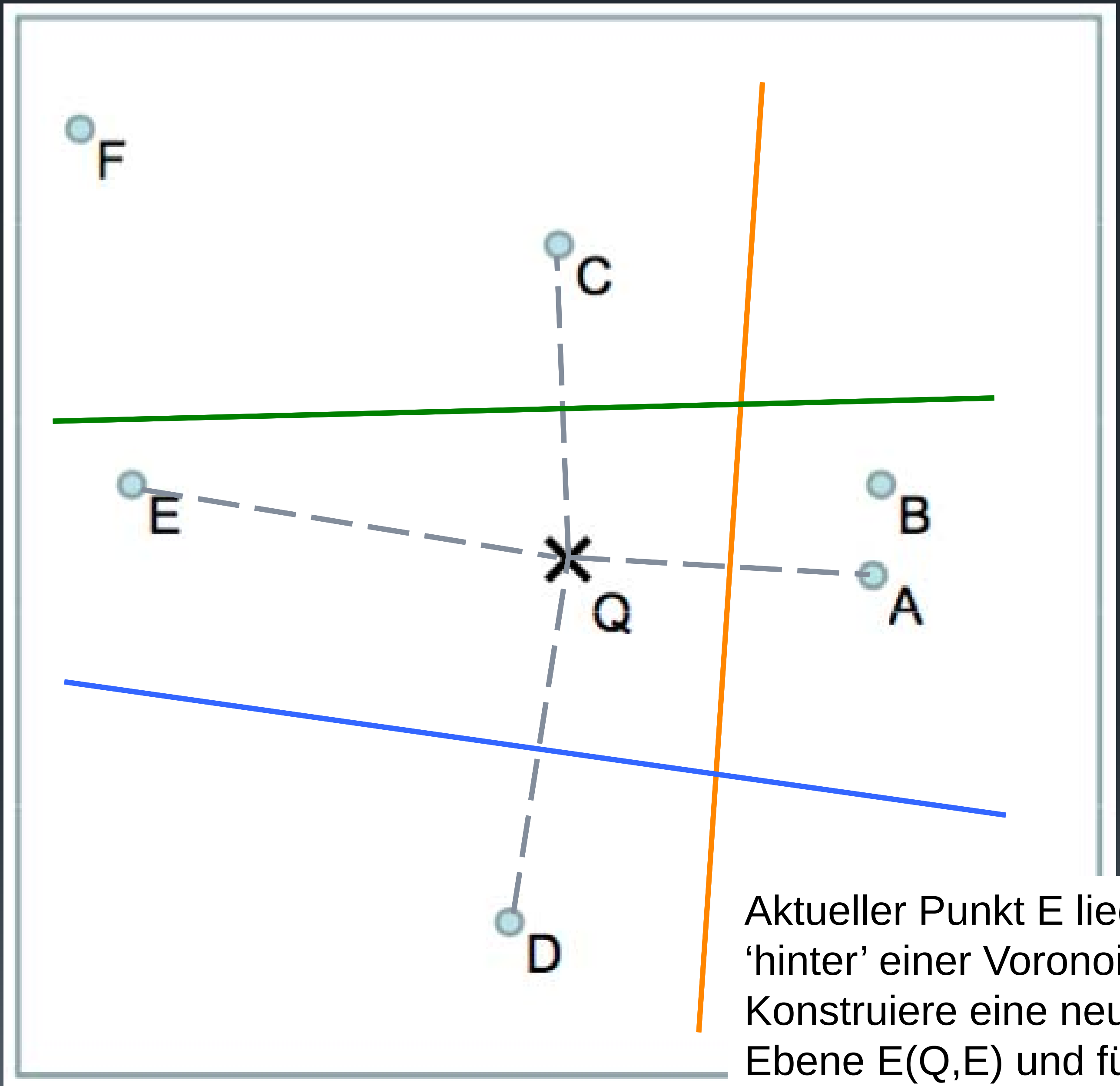


Punkte/Directoryseiten, die „hinter“ einer der Voronoi-Ebenen liegen (außer der eigenen), können aus dem Ranking/Kandidatenmenge gelöscht werden → Entferne Punkt F

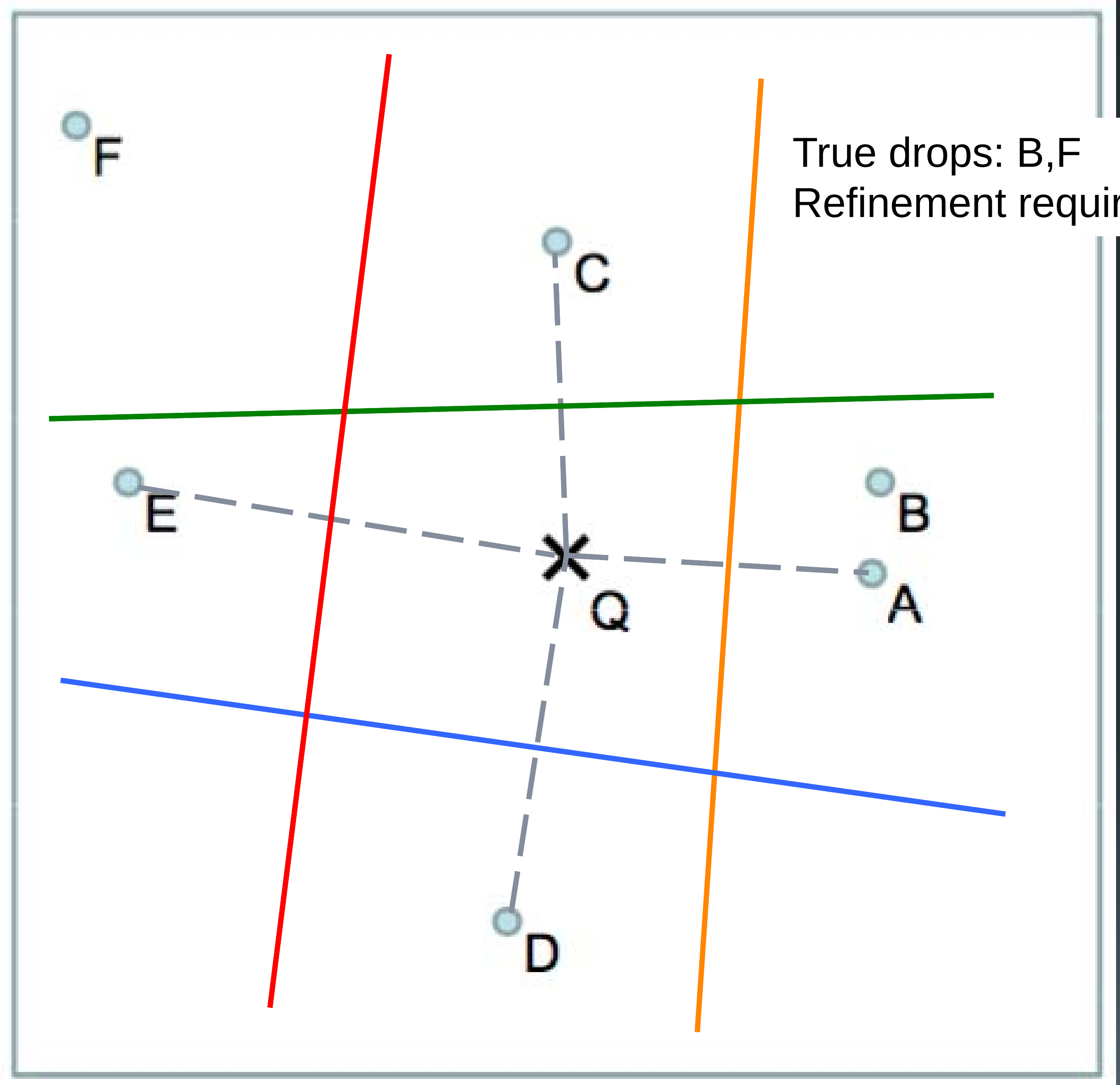


Aktueller Punkt C liegt nicht ‘hinter’ einer Voronoi-Ebene. Konstruiere eine neue Voronoi-Ebene E(Q,D) und füge D zur Kandidatenmenge hinzu

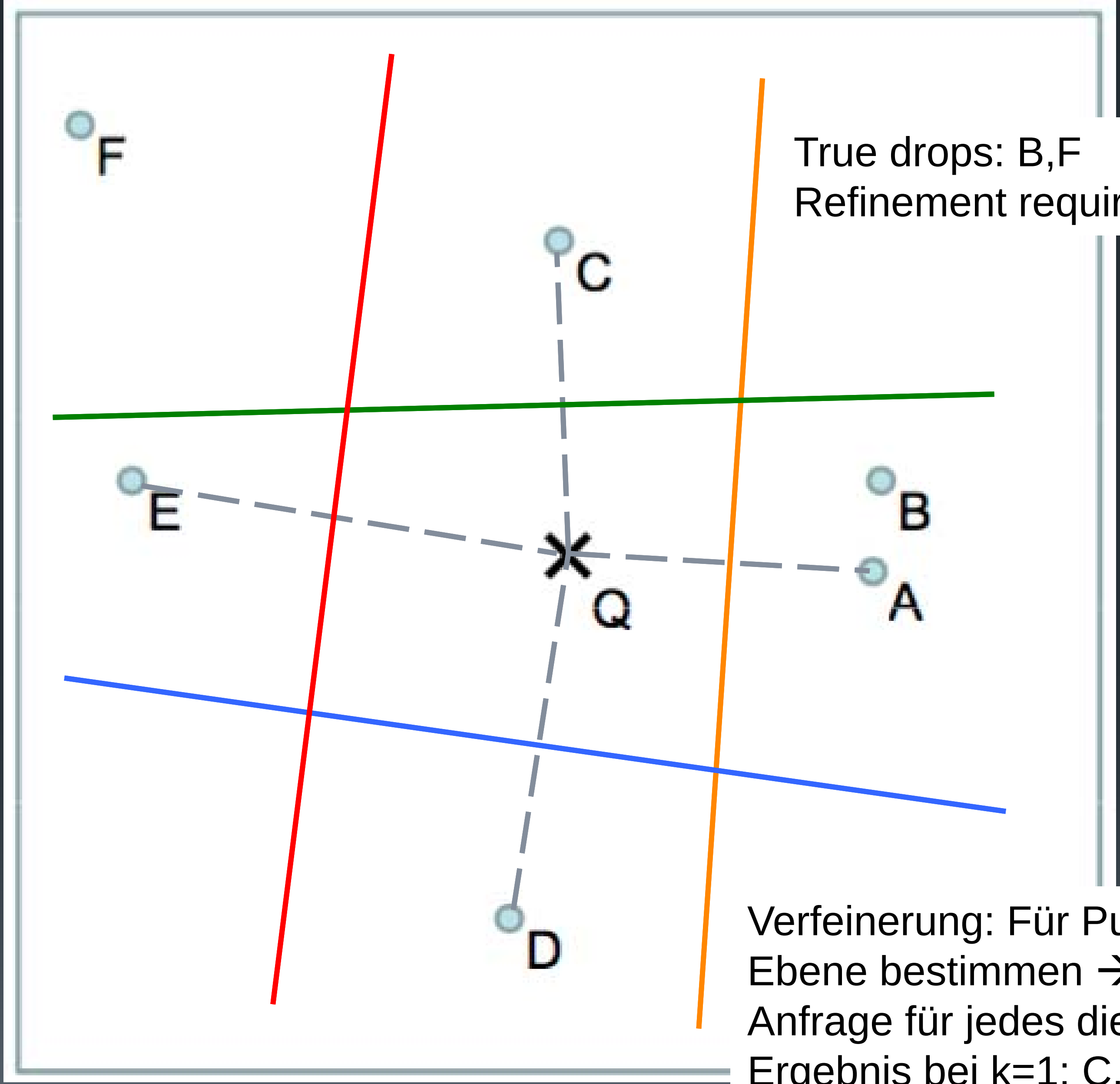

F
C
E
B
Q
A
D



Aktueller Punkt E liegt nicht ‘hinter’ einer Voronoi-Ebene. Konstruiere eine neue Voronoi-Ebene E(Q,E) und füge E zur Kandidatenmenge hinzu



True drops: B,F
Refinement required: A,C,D,E



True drops: B,F
Refinement required: A,C,D,E

Verfeinerung: Für Punkte die die Ebene bestimmen → Berechne NN Anfrage für jedes dieser Punkte
Ergebnis bei k=1: C,D