Globalübung 08

Aufgabe 8-2

DBS Prof. Dr. Thomas Seidl

Globalübung 8

- Bestimmen Sie die Zeit- und Platzkomplexität des Dijkstra-Algorithmus für einen zusammenhängenden Graphen.

Dijkstra-Algorithmus: Implementierung

S: Menge der bereits abgearbeiteten Knoten

D: aktuelle Kostenschätzung für den Pfad von v_0 zu allen anderen Knoten

Startknoten v_0, Endknoten w

$$c[v,w]\begin{cases}\geq 0 & \text{, falls eine Kante von } v \text{ nach } w \text{ existiert}\\ = \infty & \text{, falls keine Kante von } v \text{ nach } w \text{ existiert}\\ = 0 & \text{, für } w = v\end{cases}$$

```
Dijkstra(G,v_0){
  S ← {v_0}
  forall v ∈ V {
    D(v) ← c[v_0, v]
  }
  while V − S ≠ ∅ {
    w_min ← argmin_{w∈V−S} D(w)
    S ← S ∪ {w_min}
    for each v ∈ V − S {
      D(v) = min(D[v], D[w_min] + c[w_min, v])
    }
  }
}
```

Dijkstra-Algorithmus: Implementierung, Zeitkomplexität

S: Menge der bereits abgearbeiteten Knoten

D: aktuelle Kostenschätzung für den Pfad von v_0 zu allen anderen Knoten

Startknoten v_0, Endknoten w

$$c[v,w]\begin{cases} \geq 0 & \text{, falls eine Kante von } v \text{ nach } w \text{ existiert} \\ = \infty & \text{, falls keine Kante von } v \text{ nach } w \text{ existiert} \\ = 0 & \text{, für } w = v \end{cases}$$

Dijkstra(G,v_0){
  $S \leftarrow \{v_0\}$
  forall $v \in V$ {
    $D(v) \leftarrow c[v_0, v]$
  }
  while $V - S \neq \emptyset$ {
    $w_{min} \leftarrow \underset{w \in V-S}{\operatorname{argmin}} D(w)$
    $S \leftarrow S \cup \{w_{min}\}$
    for each $v \in V - S$ {
      $D(v) = \min(D[v], D[w_{min}] + c[w_{min}, v])$
    }
  }
}

Dijkstra-Algorithmus: Implementierung, Zeitkomplexität

S: Menge der bereits abgearbeiteten Knoten

D: aktuelle Kostenschätzung für den Pfad von v_0 zu allen anderen Knoten

Startknoten v_0, Endknoten w

$$c[v,w]\begin{cases}\geq 0 & \text{, falls eine Kante von } v \text{ nach } w \text{ existiert}\\ = \infty & \text{, falls keine Kante von } v \text{ nach } w \text{ existiert}\\ = 0 & \text{, für } w = v\end{cases}$$

Dijkstra(G,v_0){
  $S \leftarrow \{v_0\}$
  forall $v \in V$ {
    $D(v) \leftarrow c[v_0, v]$
  }
  while $V - S \neq \emptyset$ {
    $w_{min} \leftarrow \underset{w \in V-S}{\operatorname{argmin}} D(w)$
    $S \leftarrow S \cup \{w_{min}\}$
    for each $v \in V - S$ {
      $D(v) = \min(D[v], D[w_{min}] + c[w_{min}, v])$
    }
  }
}

O(|V|) (könnte sogar in O(|E|) gelöst werden)

O(|V|)

O(|V|)*O(|V|)

Max(O(|V|)*O(|V|), O(|V|))

Dijkstra-Algorithmus: Implementierung, Platzkomplexität

S: Menge der bereits abgearbeiteten Knoten

D: aktuelle Kostenschätzung für den Pfad von v_0 zu allen anderen Knoten

Startknoten v_0, Endknoten w

$$c[v,w]\begin{cases} \geq 0 & \text{, falls eine Kante von } v \text{ nach } w \text{ existiert} \\ = \infty & \text{, falls keine Kante von } v \text{ nach } w \text{ existiert} \\ = 0 & \text{, für } w = v \end{cases}$$

Dijkstra(G,v_0){
  $S \leftarrow \{v_0\}$ — O(1)
  forall $v \in V$ {
    $D(v) \leftarrow c[v_0, v]$
  } — O(|V|) (könnte sogar in O(|E|) gespeichert werden)
  while $V - S \neq \emptyset$ {
    $w_{min} \leftarrow \underset{w \in V-S}{\operatorname{argmin}} D(w)$ — O(1)
    $S \leftarrow S \cup \{w_{min}\}$ — O(|V|)
    for each $v \in V - S$ {
      $D(v) = \min(D[v], D[w_{min}] + c[w_{min}, v])$
    } — O(|V|)
  } — O(|V|)
}

Globalübung 8

- Bestimmen Sie die Zeit- und Platzkomplexität des Dijkstra-Algorithmus für einen zusammenhängenden Graphen.
- Bestimmen Sie die Zeit- und Platzkomplexität des Floyd-Algorithmus

Floyd-Algorithmus: Pseudo-Code

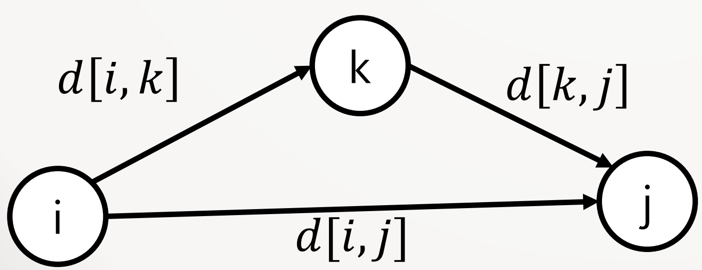
Gegeben: Kosten $c[v,w] \in \mathbb{R}_0^+ \cup \{\infty\}$ von Knoten v nach w

Für alle Knotenpaare $i, j \in \{0, \ldots, |V|-1\}$
    $d[i][j] \leftarrow c[i,j]$

Initialisierung von Matrix d:
- Jeder Knoten hat Distanz 0 zu sich selbst
- Sonst übernehmen wir erst einmal die direkten (schon bekannten) Verbindungen

Für alle $i \in \{0, \ldots, |V|-1\}$
    Für alle $j \in \{0, \ldots, |V|-1\}$
        Für alle $k \in \{0, \ldots, |V|-1\}$
            $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$

Falls Weg über k besser / kürzer als bisher bester Weg, ist dieser Weg nun der Favorit.



Floyd-Algorithmus: Pseudo-Code

Gegeben: Kosten $c[v,w] \in \mathbb{R}_0^+ \cup \{\infty\}$ von Knoten v nach w

Für alle Knotenpaare $i,j \in \{0,\dots,|V|-1\}$
    $d[i][j] \leftarrow c[i,j]$

Für alle $i \in \{0,\dots,|V|-1\}$
    Für alle $j \in \{0,\dots,|V|-1\}$
        Für alle $k \in \{0,\dots,|V|-1\}$
            $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$

Floyd-Algorithmus: Weginformationen

Gegeben: Kosten $c[v,w] \in \mathbb{R}_0^+ \cup \{\infty\}$ von Knoten v nach w

Für alle Knotenpaare $i,j \in \{0,\dots,|V|-1\}$

  $d[i][j] \leftarrow c[i,j]$

  $P[i][j] \leftarrow j$

$$\begin{pmatrix} P_{11} & \cdots & P_{1j} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{i1} & \cdots & P_{ij} & \cdots & P_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nj} & \cdots & P_{nn} \end{pmatrix}$$

Der kürzeste Weg von i nach j verläuft über P_{ij}.

Für alle $k \in \{0,\dots,|V|-1\}$

  Für alle $i \in \{0,\dots,|V|-1\}$

    Für alle $j \in \{0,\dots,|V|-1\}$

      $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$

      $P[i][j] \leftarrow k$

$\boxed{\text{$P$ speichert für zwei Knoten i,j den Knoten k, der auf dem optimalen Pfad als nächster Knoten gewählt wird.}}$

Floyd-Algorithmus: Zeitkomplexität

Gegeben: Kosten $c[v,w] \in \mathbb{R}_0^+ \cup \{\infty\}$ von Knoten v nach w

Für alle Knotenpaare $i,j \in \{0,\dots,|V|-1\}$
    $d[i][j] \leftarrow c[i,j]$
    $P[i][j] \leftarrow j$

$O(|V| * |V|)$

Für alle $k \in \{0,\dots,|V|-1\}$
    Für alle $i \in \{0,\dots,|V|-1\}$
        Für alle $j \in \{0,\dots,|V|-1\}$
            $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$
            $P[i][j] \leftarrow k$

$O(|V| * |V| * |V|)$

$O(|V|^3)$

Floyd-Algorithmus: Platzkomplexität

Gegeben: Kosten $c[v,w] \in \mathbb{R}_0^+ \cup \{\infty\}$ von Knoten v nach w

Für alle Knotenpaare $i,j \in \{0,\ldots,|V|-1\}$ — $O(2 * |V| * |V|)$

- $d[i][j] \leftarrow c[i,j]$ — $O(1)$
- $P[i][j] \leftarrow j$ — $O(1)$

$O(2)$

Für alle $k \in \{0,\ldots,|V|-1\}$

- Für alle $i \in \{0,\ldots,|V|-1\}$
  - Für alle $j \in \{0,\ldots,|V|-1\}$
    - $d[i][j] \leftarrow \min(d[i][j], d[i][k] + d[k][j])$
    - $P[i][j] \leftarrow k$

$O(2 * \ |V|^2)$

$O(|V|^2)$