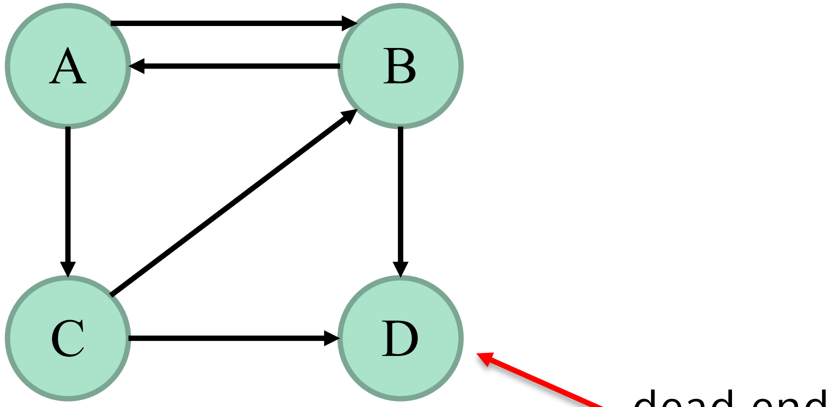
Big Data Management and Analytics
Assignment 11

Assignment 11-1

a) Explain how the PageRank algorithm avoids to get stuck in a „dead end“.

What is a „dead end“?

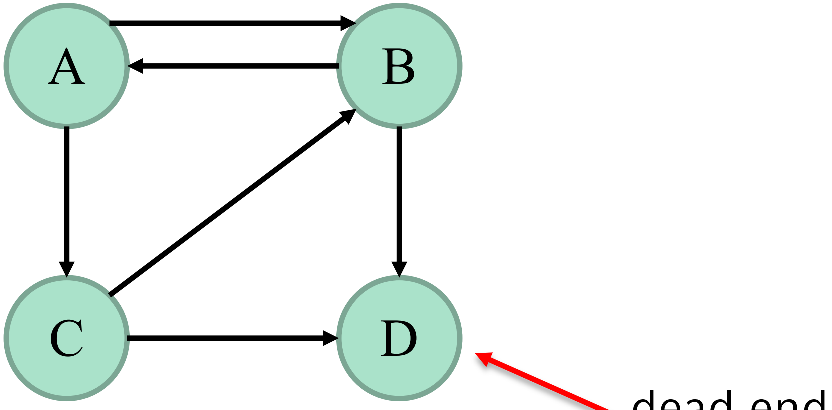
→ Pages which do not have any outgoing link



Assignment 11-1

a) Explain how the PageRank algorithm avoids to get stuck in a „dead end".

- PageRank avoids getting trapped in such „dead ends" by randomly surfing on any other page → Teleport
- Let n be the number of pages in the graph. The probability to surf on a specific page is $^1/_n$
- In the graph below, being trapped in D, the next pages A,B,C or D itself can be visited with a probability of $^1/_4$

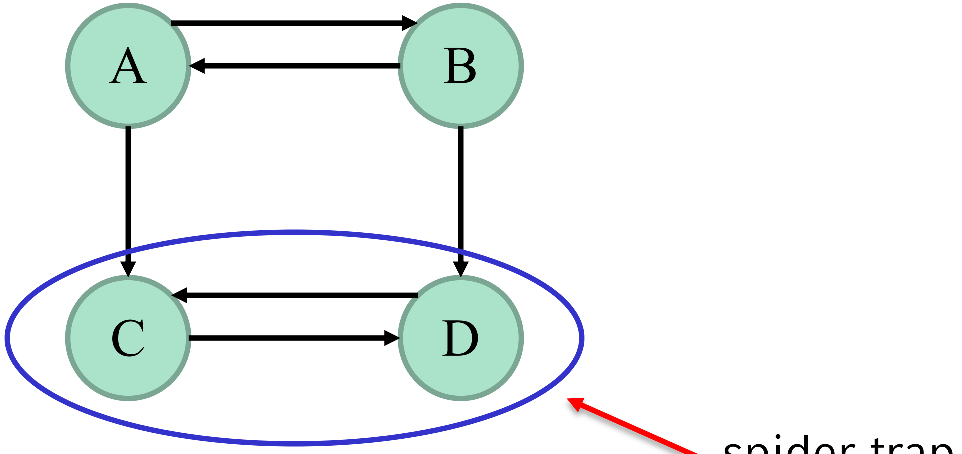


Assignment 11-1

a) Explain how the PageRank algorithm avoids to get stuck in a „dead end“.

What is a „spider trap“?

→ All outgoing links of a group of pages are **within** the group



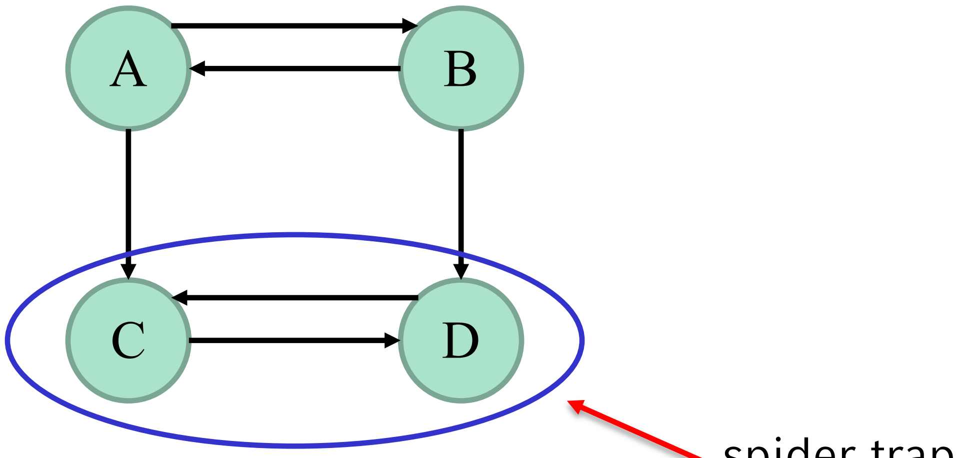
Assignment 11-1

a) Explain how the PageRank algorithm avoids to get stuck in a „dead end“.

How can a spider trap being avoided?

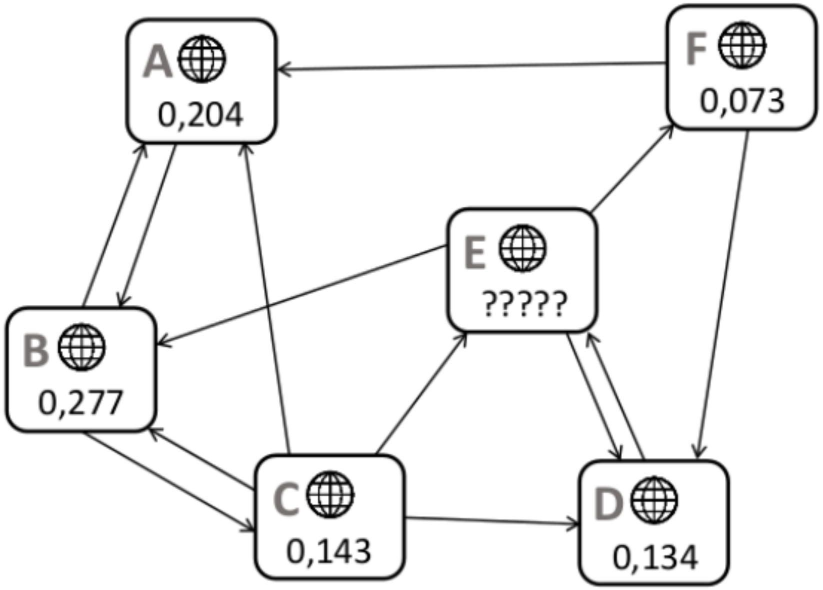
→ Follow a link with a probability of β

→ Teleporting randomly to any page is done with a probability of $(1-\beta)$



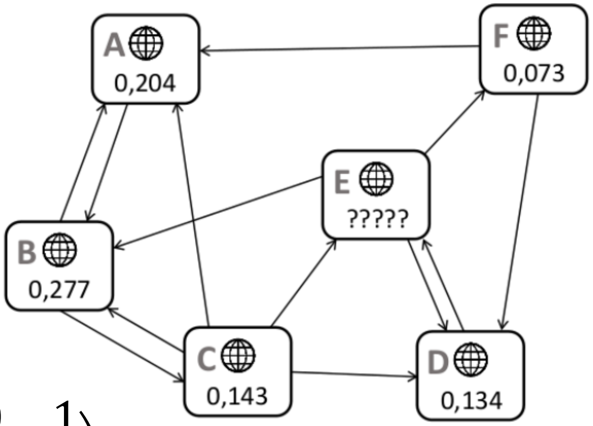
Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$



Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$



1. Create the adjacency matrix A and A^T

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad A^T = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$

2. Create a matrix M by norming A^T

$$A^T = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathrm{M} = \begin{pmatrix} 0 & 1/2 & 1/4 & 0 & 0 & 1/2 \\ 1 & 0 & 1/4 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 1/3 & 1/2 \\ 0 & 0 & 1/4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 \end{pmatrix}$$

Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$

2. Create a matrix M by norming A^T

$$M = \begin{pmatrix} 0 & 1/2 & 1/4 & 0 & 0 & 1/2 \\ 1 & 0 & 1/4 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 1/3 & 1/2 \\ 0 & 0 & 1/4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 \end{pmatrix}$$

- This matrix is alread real, non-negative and its columns fulfill the stochastic property
- In case of a dead end these properties would not hold

Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$

3. Compute the Google matrix $\boxed{G = \beta \cdot M + \frac{1-\beta}{n} \cdot \mathbf{1}}$

$$0.85 \cdot \begin{pmatrix} 0 & 1/2 & 1/4 & 0 & 0 & 1/2 \\ 1 & 0 & 1/4 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/4 & 0 & 1/3 & 1/2 \\ 0 & 0 & 1/4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 \end{pmatrix} + (1 - 0.85) \begin{pmatrix} 1/6 & \cdots & 1/6 \\ \vdots & \ddots & \vdots \\ 1/6 & \cdots & 1/6 \end{pmatrix} =$$

Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$

3. Compute the Google matrix $\boxed{G = \beta \cdot M + \frac{1-\beta}{n} \cdot \mathbf{1}}$

$$= \frac{1}{60} \cdot \begin{pmatrix} 0 & 25.5 & 12.75 & 0 & 0 & 25.5 \\ 51 & 0 & 12.75 & 0 & 17 & 0 \\ 0 & 25.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12.75 & 0 & 17 & 25.5 \\ 0 & 0 & 12.75 & 51 & 0 & 0 \\ 0 & 0 & 0 & 0 & 17 & 0 \end{pmatrix} + \frac{1}{60} \begin{pmatrix} 1.5 & \cdots & 1.5 \\ \vdots & \ddots & \vdots \\ 1.5 & \cdots & 1.5 \end{pmatrix} =$$

Assignment 11-1

b) Given the graph below, compute the Google Matrix with $\beta = 0.85$

3. Compute the Google matrix $\boxed{G = \beta \cdot M + \frac{1-\beta}{n} \cdot \mathbf{1}}$

$$= \frac{1}{60} \cdot \begin{pmatrix} 1.5 & 27 & 14.25 & 1.5 & 1.5 & 27 \\ 52.5 & 1.5 & 14.25 & 1.5 & 18.5 & 1.5 \\ 1.5 & 27 & 1.5 & 1.5 & 1.5 & 1.5 \\ 1.5 & 1.5 & 14.25 & 1.5 & 18.5 & 27 \\ 1.5 & 1.5 & 14.25 & 52.5 & 1.5 & 1.5 \\ 1.5 & 1.5 & 1.5 & 1.5 & 18.5 & 1.5 \end{pmatrix}$$

Assignment 11-1

c) How can the actual PageRank values be computed by using the Google Matrix?

- The Google matrix has the stochastic property
  → There exists an eigenvector of G with the eigenvalue 1

- Looking at the eigenvalue problem $G \cdot x = x$,
  vector x is a stochastic vector which consists of the PageRank values

- For getting the eigenvector x_i to the corresponding eigenvalue λ_i
  we can solve the following equation system: $(G\boldsymbol{x} - \lambda_i \boldsymbol{E}x) = 0$

- In our case $\lambda_i = 1$, thus we need to compute $(G\boldsymbol{x} - \boldsymbol{E}\boldsymbol{x}) = 0$
  in order to get the eigenvector

Assignment 11-1

c) How can the actual PageRank values be computed by using the Google Matrix?

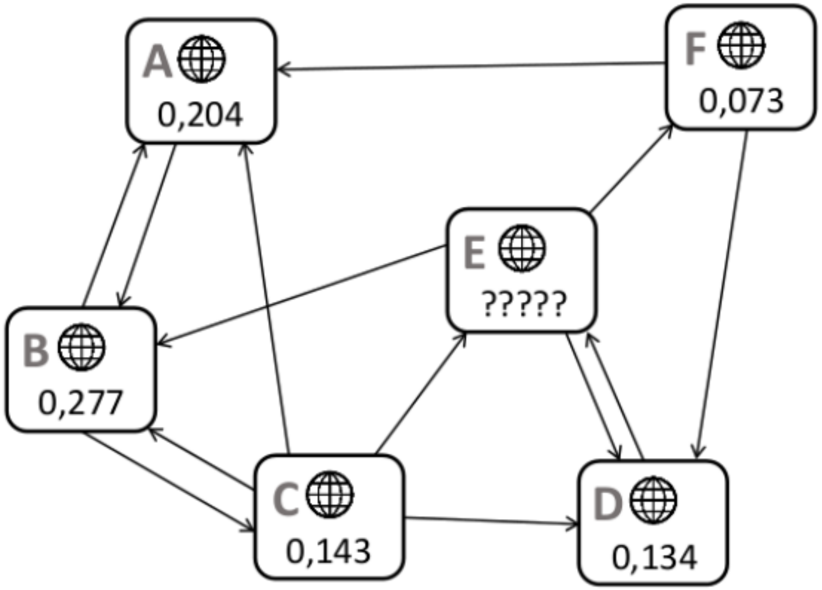
- Solve the following equation system $= (\boldsymbol{G}\boldsymbol{x} - \boldsymbol{E}\boldsymbol{x})0$:

```
x = np.repeat(1./6., 6)
x_prev = x
for i in range(1000):
    print(f'iteration {i}:',x)
    x = M.dot(x)
    x = x/np.linalg.norm(x,1)
    if np.allclose(x, x_prev):
        break
    x_prev = x
```

```
iteration 0: [0.16666667 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667]
iteration 1: [0.20208333 0.24930556 0.09583333 0.17847222 0.20208333 0.07222222]
iteration 2: [0.18201389 0.27439236 0.13095486 0.13331597 0.19706597 0.08225694]
iteration 3: [0.20440386 0.26337507 0.14161675 0.14362247 0.16614648 0.08083536]
iteration 4: [0.20138299 0.27591168 0.13693441 0.13652342 0.17717266 0.07207484]
iteration 5: [0.20199283 0.27547303 0.14226246 0.13492929 0.17014347 0.07519892]
iteration 6: [0.20426635 0.275132   0.14207604 0.13539763 0.16992067 0.07320732]
iteration 7: [0.20323537 0.27696174 0.1419311  0.13444846 0.17027914 0.07314419]
iteration 8: [0.20395538 0.27615618 0.14270874 0.1344924  0.16944155 0.07324576]
iteration 9: [0.20382143 0.27669612 0.14236638 0.13446349 0.16964414 0.07300844]
iteration 10: [0.20387729 0.27656691 0.14259585 0.13434728 0.16954682 0.07306584]
iteration 11: [0.20389554 0.27663558 0.14254094 0.13439287 0.16949681 0.07303827]
iteration 12: [0.20390134 0.27662525 0.14257012 0.13435531 0.16952389 0.0730241 ]
iteration 13: [0.20389712 0.27664405 0.14256573 0.13436316 0.16949816 0.07303177]
iteration 14: [0.20390744 0.27663225 0.14257372 0.1343582  0.1695039  0.07302448]
iteration 15: [0.20390103 0.27664435 0.14256871 0.13435843 0.16950139 0.07302611]
iteration 16: [0.20390579 0.27663712 0.14257385 0.13435734 0.16950051 0.07302539]
iteration 17: [0.20390351 0.27664201 0.14257077 0.13435788 0.16950068 0.07302515]
iteration 18: [0.20390483 0.27663946 0.14257286 0.13435717 0.16950049 0.07302519]
```

Assignment 11-1

d) Given the graph as seen below, compute the missing PageRank value of node E by using the PageRank equation



d) Given the graph as seen below, compute the missing PageRank value of node E by using the PageRank equation

$$\boxed{PR_E = \frac{1-\beta}{n} + \beta \sum_{i \to E} \frac{PR_i}{d_i}}$$

$$PR_E = \frac{0.15}{6} + 0.85(\frac{0.143}{4} + 0.134)$$

$$PR_E = 0.169$$