Spletne tehnologije

VAJA 3

Niko Lukač

VAJA 3

V okviru dane vaje boste preučili izbran spletni graf ter uvrstili spletne strani glede na njihovo povezanost z uporabo algoritma PageRank. Za realizacijo vaje lahko uporabite poljubni programski jezik in tehnologije.

Zahteve

1. Najprej implementirajte preprost spletni pajek (angl. web crawler), ki se poveže na določeno spletno stran (vrhvno pot "/"), ter na vse ostale spletne strani (samo vrhovne poti "/"), ki so vsebovane v hiperpovezavah. Dani spletni pajek se tako pretaka po več spletnih straneh do **določene globine**, ki je uporabniško nastavljiva.
    - Hranite seznam že obiskanih spletnih strani.

```
from bs4 import BeautifulSoup
import requests
import re

url = "https://google.com"
h = response = requests.get(url).text
soup = BeautifulSoup(h, 'html.parser')

for link in soup.find_all('a', attrs={'href': re.compile("^https://")}):
    print(link.get('href'))
```

Zahteve

- Pri tem najprej preverite ali je dovoljeno obiskat stran s strani spletnega pajka preko datoteke robots.txt, kjer preverite, da se pot "/" ne nahaja v "Disallow: " oz. je v "Allow: ", oz. ni pravila. Za implementacijo pajka lahko uporabite knjižnice. (3%)

Primer robots.txt:

```
User-agent: Googlebot
Disallow: /nogooglebot/

User-agent: *
Allow: /

Sitemap: https://www.example.com/sitemap.xml
```

Zahteve

2. Nad obiskanimi spletnimi stranmi iz spletnega pajka zgradite usmerjeni spletni graf. Vozlišče v grafu je spletna stran, povezava med dvema vozliščama je hiperpovezava. Nad spletnim grafom izračunate metriko PageRank (s teleportiranjem) za vsa vozlišča. Za implementacijo algoritma PageRank ne uporabite knjižnic. Uporabite parameter **Beta=0.85**, izračun pa naj poteka do **konvergence** (zelo majhne spremembe v vrednostih). (3%)

$$A = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

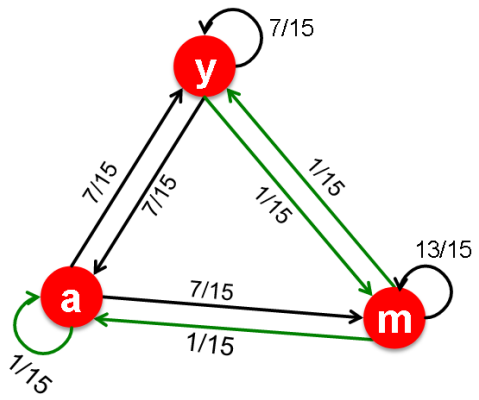
$$\boldsymbol{r}^{\text{new}} = \boldsymbol{A} \cdot \boldsymbol{r}^{\text{old}}$$

Zahteve

2. Nad obiskanimi spletnimi stranmi iz spletnega pajka zgradite usmerjeni spletni graf. Vozlišče v grafu je spletna stran, povezava med dvema vozliščama je hiperpovezava. Nad spletnim grafom izračunate metriko PageRank (s teleportiranjem) za vsa vozlišča. Za implementacijo algoritma PageRank ne uporabite knjižnic. Uporabite parameter **Beta=0.85**, izračun pa naj poteka do **konvergence** (zelo majhne spremembe v vrednostih). (3%)

$$A = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

$$r^{new} = A \cdot r^{old}$$



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

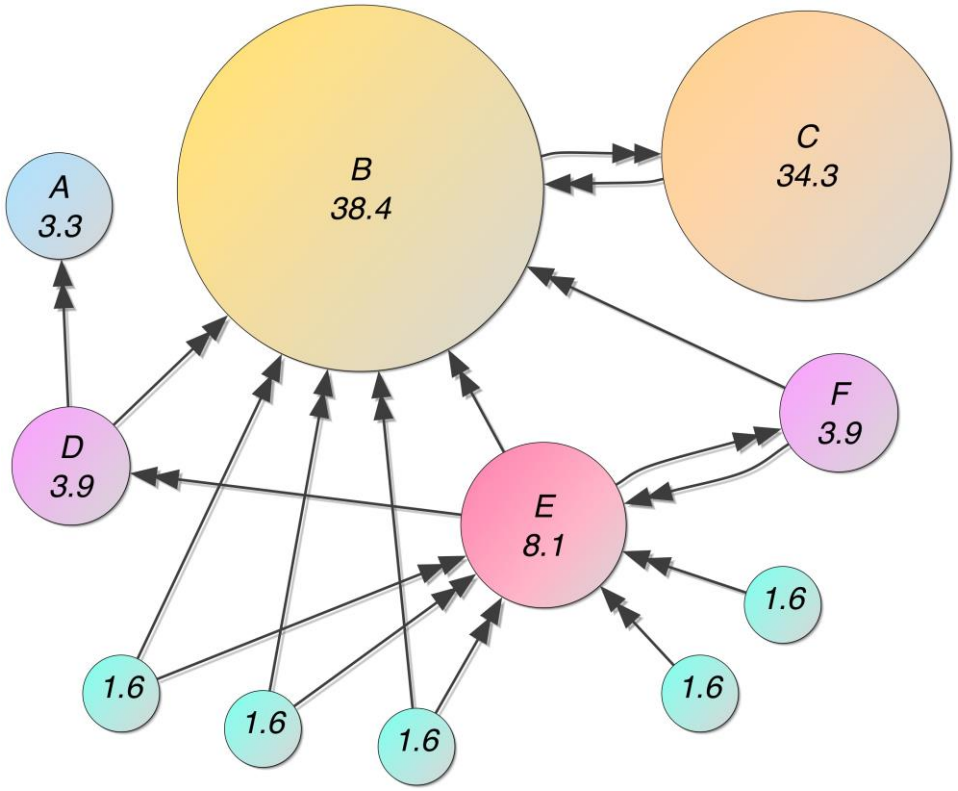
M $\qquad$ $[1/N]_{NxN}$

$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{ccccc} 1/3 & 0.33 & 0.24 & 0.26 & \\ 1/3 & 0.20 & 0.20 & 0.18 & \ldots \\ 1/3 & 0.46 & 0.52 & 0.56 & \end{array} \begin{array}{c} \\ 7/33 \\ 5/33 \\ 21/33 \end{array}$$

Zahteve

3. Izdelajte poročilo (1-2 strani), kjer izrišete dva poljubna spletna grafa (lahko programersko ali ročno npr. z draw.io) ter izpišete prvih 5 spletnih strani po metriki PageRank (ter podajte izračune) za vsak spletni graf. Globina preiskovanja s pajkom naj bo minimalno 3 in maksimalno 6. Velikost vozlišč naj bo relativno proporcionalna ranking-u. (7%)



B 38.4
C 34.3
E 8.1
F 3.9
D 3.9
A 3.3

Zahteve

- **Oddajte naslednje v arhivu ZIP:** programska koda za crawler in izračun PageRank-a, porocilo.pdf

- **Vrednost naloge:** 13% od celotne ocene pri predmetu
- **Naknadni roki za oddajo:** 2 tedna pred vsakim izpitnim rokom (datumi bodo objavljeni sproti na uvodni strani predmeta)
- Oddana vaja bo ocenjena najkasneje v 2 tednih po roku za oddajo ali v 1 tednu pred izpitnim rokom. Ustnega zagovora pri dani vaji ni. V primeru nestrinjanja z dodeljeno oceno je možen ustni zagovor po predhodnem dogovoru.

- **Literatura:**

  - robots.txt: https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt
  - Python BeautifulSoup: https://www.scrapingbee.com/blog/crawling-python/
  - Python networkx: https://medium.com/@nelsonjoseph123/graph-visualization-using-python-bbd9a593c533