# Eksploracja danych

## Projekt 4: Przeszukiwanie sieci i algorytm HITS

Celem tego zadania programistycznego jest zaimplementowanie algorytmu HITS do obliczenia wartości Hub oraz Authority zbioru stron internetowych związanych z określonym tematem. Te informacje można wykorzystać do zdecydowania, które strony w zestawie można uznać za najbardziej kompetentne w tej dziedzinie (czyli mają największą wagę „autorytetu"), a które być uważane za najlepszy w linkowaniu do dobrych stron w danej dziedzinie (czyli mają największą wagę „hubowatości").

Załóżmy, że plik wejściowy o nazwie links.txt składa się z tekstu ASCII, który wygląda następująco:

```
http://www.it.uu.se/index.html http://www.uu.se http://www.kth.se/

http://www.kth.se www.uu.se http://www.slu.se

http://www.uu.se http://www.it.uu.se www.dn.se

http://user.it.uu.se/~pergu http://www.uu.se http://www.it.uu.se/

http://www.slu.se
```

Każdy wiersz można interpretować w następujący sposób:

Pierwsza strona internetowa w każdym wierszu zawiera linki do pozostałych stron internetowych w tym wierszu. Informacje nie są kompletne: niektóre strony internetowe, do których prowadzą łącza z innej witryny, nie pojawiają się na początku wiersza. Niektóre wiersze prawdopodobnie zawierają tylko jeden adres internetowy (jak ostatni w powyższym przykładzie), co oznacza, że ta strona internetowa nie zawiera linków do żadnej innej strony internetowej.

Zwróć uwagę, że niektóre linki pojawiają się w różnych formach, np.

- `http://www.it.uu.se/index.html`
- `http://www.it.uu.se/`
- `http://www.it.uu.se`

Wszystkie prowadzą do tej samej strony, mimo że wyglądają inaczej. Innym przykładem jest:

- `www.uu.se`
- `http://www.uu.se`

Jeden z linków jest poprzedzony przedrostkiem „`http //`", a drugi nie. Nadal prowadzą do tej samej strony. Ponieważ plik tekstowy może zawierać te różne typy linków, musisz wstępnie przetworzyć plik, aby to zrobić. Wszystkie linki prowadzące do tej samej strony mają tę samą nazwę.

Twoim zadaniem jest napisanie programu w MatLabie, który oblicza wagę „hubowatości" i wagę autorytetu każdej strony internetowej w pliku. Jedyny parametr, jaki musi przyjąć Twój program, to nazwa pliku. (Jeśli używasz iteracyjnego algorytmu do obliczania wag, możesz również pozwolić, aby liczba iteracji była parametrem wejściowym). Program powinien zwracać 10 najbardziej autorytatywnych stron w formacie zbioru wraz z ich wagami „autorytetu", a także 10 najbardziej „hubowatych" stron z ich wagami „hubowatości." Oto przykład:

```
> myHITS linkis.txt

Strony o wysokim "autorytecie"

=========================================

Strona                Waga autorytetu
```

```
========================================

http://www.uu.se            0.8363

http://www.it.uu.se         0.3954

http://www.kth.se           0.2685

http://www.slu.se           0.2685

http://user.it.uu.se/~pergu 0.0000

Strony o dużej "hubowatości"

========================================

Strona               Waga "hubowatości"

========================================

http://user.it.uu.se/~pergu 0.6072

http://www.it.uu.se         0.5446

http://www.kth.se           0.5446

http://www.uu.se            0.1949

http://www.slu.se           0.0000
```

Twój program musi najpierw wstępnie przetworzyć pliki, aby rozpoznać nazwy linków, do których prowadzą różne linki oraz usunąć wszystkie linki, dla których nie ma informacji o tym, do czego prowadzą.

Aby przetestować swoją implementację, otrzymasz cztery różne pliki w archiwum:

`http://www.math.us.edu.pl/~pgladki/teaching/2021-2022/ed-p05.zip`.

Każdy plik zawiera strony internetowe na określony temat. Przeanalizuj wyniki swojego programu, aby dowiedzieć się, czy strony o wysokiej wadze „hubowatości” lub „autorytetu” faktycznie są „hubowate” lub „autorytatywne”.