



Министерство науки и высшего образования  
Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана (национальный  
исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

Факультет «Информатика и системы управления»

## ОТЧЕТ по лабораторной работе №5

по дисциплине

«Информационный поиск и извлечение информации из  
текстов»

Студент группы ИУ9-21М

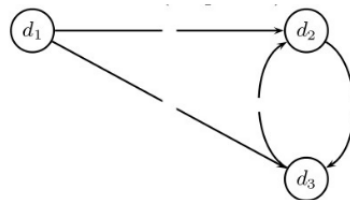
\_\_\_\_\_ С.С. Погосян  
(подпись, дата)

Руководитель

\_\_\_\_\_ Н.В. Лукашевич  
(подпись, дата)

## 1. Постановка задачи

### Задача



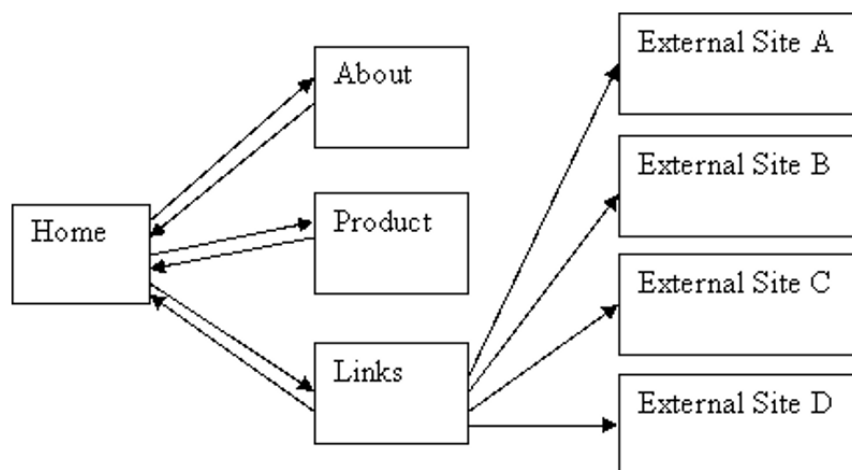
Дан веб-граф, составить матрицу переходов.

Коэффициент телепортации = 0.1.

Составить матрицу переходов и вычислить pagerank для узлов сети

Начальный вектор состояний можно

взять с равными вероятностями для каждого состояния



## 2. Решение

Приведенный в приложении код (доступен по ссылке: <https://github.com/legion15q/sem2/tree/master/num5/py>) вычисляет pagerank документов для полученных матриц перехода каждого веб-графа.

Матрица переходов для веб-графа 1:

$$A_1 = \begin{pmatrix} \frac{1}{30} & \frac{1}{30} + 0.9 * \frac{1}{2} & \frac{1}{30} + 0.9 * \frac{1}{2} \\ \frac{1}{30} & \frac{1}{30} & \frac{1}{30} + 0.9 \\ \frac{1}{30} & \frac{1}{30} + 0.9 & \frac{1}{30} \end{pmatrix}.$$

Начальное вектор состояния будет заполнен равными вероятностями оказаться в том или ином состоянии:

$$x_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}).$$

Аналогично для веб-графа 2:

$$A_2 = \begin{pmatrix} 0.0125 & 0.3125 & 0.3125 & 0.3125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 \\ 0.9125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 \\ 0.9125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 & 0.0125 \\ 0.1925 & 0.0125 & 0.0125 & 0.0125 & 0.1925 & 0.1925 & 0.1925 & 0.1925 \\ 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \\ 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \\ 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \\ 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \end{pmatrix},$$

$$x_2 = (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}).$$

Критерий останова выберем следующий:

$$|xA - x_{i+1}A| < \epsilon,$$

где  $\epsilon = 0.0001$  Итерационный процесс сходится для каждой матрицы со следующими параметрами:

1.  $i = 2, \quad x = (0.03333333, 0.48333333, 0.48333333)$
2.  $i = 18, \quad x = (0.3148, 0.1380, 0.1380, 0.1380, 0.0677, 0.0677, 0.0677, 0.0677)$

### 3. Приложение

```
import numpy as np
```

```
def main():
    x = [1 / 3, 1 / 3, 1 / 3]
    A = ([0.033, 0.483, 0.483],
         [0.033, 0.033, 0.933],
         [0.033, 0.933, 0.033])
```

```
)
eps = [1e-3]*len(x)
i = 0
while True:
    x_new = np.dot(x, A)
    i += 1
    if all(np.dot(x, A) - np.dot(x_new, A) < eps):
        break
    x = x_new
print(i)
print(x_new)

A = ([0.0125, 0.3125, 0.3125, 0.3125, 0.0125, 0.0125, 0.0125, 0.0125],
      [0.9125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125],
      [0.9125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125, 0.0125],
      [0.1925, 0.0125, 0.0125, 0.0125, 0.1925, 0.1925, 0.1925, 0.1925],
      [0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250],
      [0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250],
      [0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250],
      [0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250, 0.1250],
      )
x = [1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8, 1/8]
i = 0
eps = [1e-3] * len(x)
while True:
    x_new = np.dot(x, A)
    i += 1
    if all(np.dot(x, A) - np.dot(x_new, A) < eps):
        break
    x = x_new
print(i)
print(x_new)

if __name__ == '__main__':
    main()
```