

# Spis treści

<b>1</b>	<b>Macierze</b>	<b>3</b>
1.1	Tworzenie macierzy na potrzeby algorytmów . . . . .	3
1.2	Operacje przeprowadzane na macierzach . . . . .	3
1.3	Inne metody . . . . .	3

# List of Listings

# Rozdział 1

## Macierze

### 1.1 Tworzenie macierzy na potrzeby algorytmów

W algorytmach potrzebne są głównie 3 rodzaje macierzy i ich transpozycje. Są to macierze:

M użytkownicy tagi M dokumenty tagi M użytkownicy dokumenty

Wykorzystywane są one w kolejnych iteracjach algorytmu Social PageRank. Przy algorytmie Adapted PageRank również są one używane pośrednio. Struktura na której operuje algorytm Adapted pagerank jest macierzą złożoną z macierzy  $M_{UD}$ ,  $M_{TD}$ ,  $M_{UT}$  i ich transpozycji. Macierzy używana w algorytmie wygląda następująco:

$$G_f = \begin{pmatrix} 0 & M_{DU} & M_{TD}^T \\ M_{DU}^T & 0 & M_{UT} \\ M_{TD} & M_{UT}^T & 0 \end{pmatrix}$$

### 1.2 Operacje przeprowadzane na macierzach

W każdej iteracji algorytmów główną operacją przeprowadzaną jest mnożenie wymienionych wcześniej macierzy przez wektor. Operacja ta jest przeprowadzana do czasu uzyskania zbieżności wartości wektora wynikowego.

Z powodu wielkości macierzy w aplikacji nie możemy wczytać bezpośrednio całych macierzy do pamięci i na nich operować. Dodatkowo używana biblioteka stawia ograniczenie na iloczyn kolumn i wierszy takie że:  $ilosc\_kolumn * ilosc\_wierszy \leq 2^{31} - 1$ . Gdzie wartość  $2^{31} - 1$  jest to maksymalna liczba jaką można przypisać zmiennej typu integer w języku Java.

### 1.3 Inne metody