Sprawozdanie trzecie

Algorytmy grafowe

Piotr Tylczyński

20.04.2019

Spis

1 Reprezentacja grafu przez macierz sąsiedztwa **2**
1.1 Opis 2
1.2 Złożonośc podstawowych operacji 2
1.2.1 Sprawdzenie istnienia krawędzi 2
1.2.2 Znalezienie wszystkich nastepników 2
1.3 Sortowanie topologiczne 3
1.3.1 Sorotowanie za pomocą algorytmu DFS 3
1.3.2 Sortowanie za pomoca algorytmu BFS 3

2 Reprezentacja grafu przez listę krawędzi **4**
2.1 Opis 4
2.2 Złożoność podstawowych operacji 4
2.2.1 Sprawdzenie istnienia krawędzi 4
2.3 Znalezienie wszytkich następników 4
2.4 Sortowanie topologiczne 4
2.4.1 Sortowanie za pomocą algorytmu DFS 5
2.4.2 Sortowanie za pomocą algorytmu BFS 5

3 Reprezentacja grafu przez listę następników **6**
3.1 Opis 6
3.2 Złożoność podstawowych operacji 6
3.2.1 Sprawdzanie istnienia krawędzi 6
3.2.2 Znalezienie wszystkich następników 6
3.3 Sortowanie topologiczne 6
3.3.1 Sortowanie za pomocą algorytmu DFS 7
3.3.2 Sortowanie za pomocą algorytmu BFS 7

Oznaczenia **8**

1 Reprezentacja grafu przez macierz sąsiedztwa

1.1 Opis

W tej reprezentacji G jest opisywany przez zero-jedynkową macierz o n kolumnach i n wierszach. W G istnieje łuk między wierzchołkami A i B, jeżeli istnieje jedynka w macierzy sąsiedztwa znajdująca się w A kolumnie i B wierszu.

1.2 Złożonośc podstawowych operacji

1.2.1 Sprawdzenie istnienia krawędzi

Posiada złożoność:

$$O(1) \tag{1}$$

Złozonośc jest stała, ponieważ sprawdzenie istnienia krawędzi sprowadza się do sprawdzenia, czy w odpowiednim miejscu macierzy znajduje się jedynka. Odczytanie pola elementu z tabeli posiada stałą złożoność czasową, więc i operacja sprawdzania istnienia krawędzi jest stała

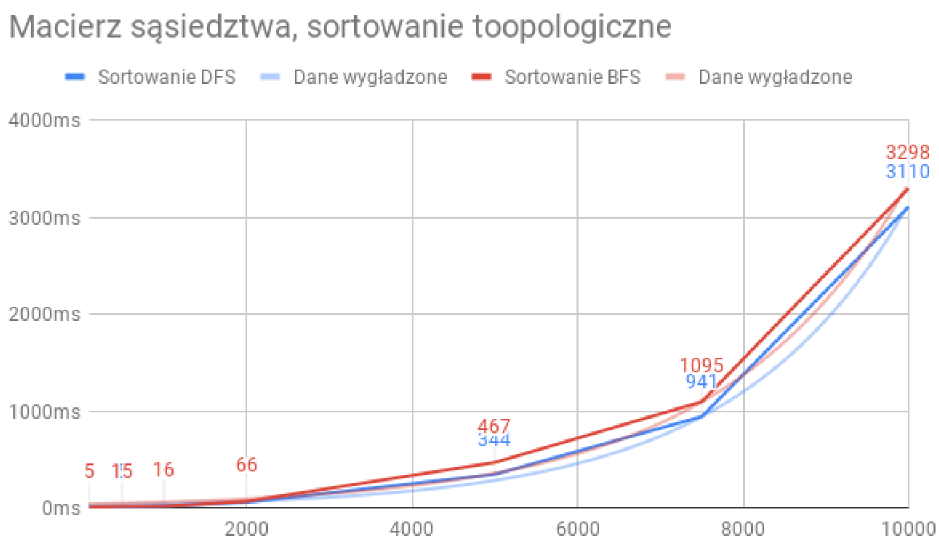
1.2.2 Znalezienie wszystkich nastepników

Posiada złożność

$$O(n) \tag{2}$$

Znalezienie nastepników w macierzy sąsiedztwa sprowadza sie od przejrzenia odpowiedniej kolumny w macierzy sąsiedztwa. Jak wiadomo macierz sąsiedztwa posiada dokładnie n elementów w kolumnie, a złożoność odczytania jednego elementu jest stała, więc odczytanie n elementów zajmie n czasu.

1.3 Sortowanie topologiczne



1.3.1 Sorotowanie za pomocą algorytmu DFS

Charakteryzuje się złożonością:

$$\text{ilość operacji szukania następników} * n = O(n) * n = O(n^2) \quad (3)$$

1.3.2 Sortowanie za pomoca algorytmu BFS

Cechuje się złożonością:

$$\text{ilość operacji szukania następników} * n = O(n) * n = O(n^2) \quad (4)$$

2 Reprezentacja grafu przez listę krawędzi

2.1 Opis

G jest w takiej reprezentacji przechowywany jako lista par uporządkowanych. W G istnieje łuk z A od B wtedy i tylko wtedy, gdy istnieje dwójka (A, B), w liście krawędzi

2.2 Złożoność podstawowych operacji

2.2.1 Sprawdzenie istnienia krawędzi

Posiada złożonośc:

$$O(m) \tag{5}$$

Wynika ona z potrzeby przejżenia całej listy w celu sprawdzenia czy istnieje w niej odpowiednia dwójka.

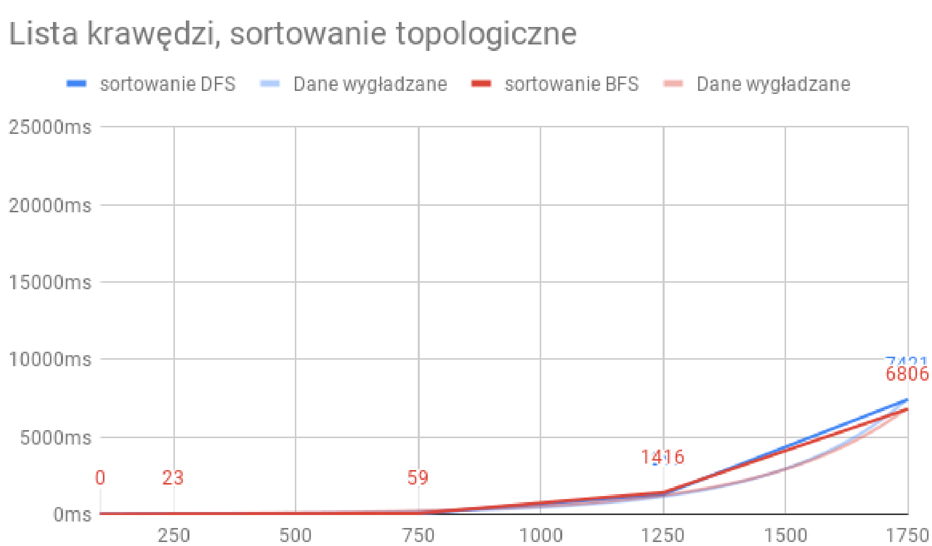
2.3 Znalezienie wszytkich następników

Posiada złożoność:

$$O(m) \tag{6}$$

Tak jak w poprzednim przypadku wynika ona z potrzeby sprawdzenia całej listy w poszukiwaniu odpowiednich dwójek.

2.4 Sortowanie topologiczne



2.4.1 Sortowanie za pomocą algorytmu DFS

Cechuje się złożonością
$$O(n^3) \tag{7}$$
która wynika z:

$$\begin{aligned}\text{ilość operacji} = \text{ilośc operacji na poszukanie wszystkich następników} * n \\ \approx m * n \end{aligned} \tag{8}$$

W testownym przypdaku, czyli gdy nasycenie G wynosi 50% ilość krawędzi można wyrazić jako:
$$m = \frac{(n)(n-1)}{4} \tag{9}$$
Co po podstawieniu do równania 8 daje wynik:

$$\text{ilość operacji} \approx \frac{(n)(n-1)}{4} * n \approx n^3 \tag{10}$$

2.4.2 Sortowanie za pomocą algorytmu BFS

Posiada złożonośc:
$$O(n^3) \tag{11}$$
Powstała ona z:

$$\text{ilość operacji} = \text{ilość operacji na znalezienie następników} * n \tag{12}$$

Dalszy tok rozumowania jest taki sam

3 Reprezentacja grafu przez listę następników

3.1 Opis

G jest w takim przypadku przechowywany jako lista, której elementami są listy następników danego wierzchołka. W takim wierzchołku istnieje łuk z A do B jeżeli istnieje lista A, której elementem jest B

3.2 Złożoność podstawowych operacji

3.2.1 Sprawdzanie istnienia krawędzi

Taka operacja ma złożoność:

$$O(n) \tag{13}$$

co wynika z faktu, że należy przeszukać tylko jedną listę, jednak w najgorszym przypadku ta lista może składać się ze wszystkich wierzchołków G

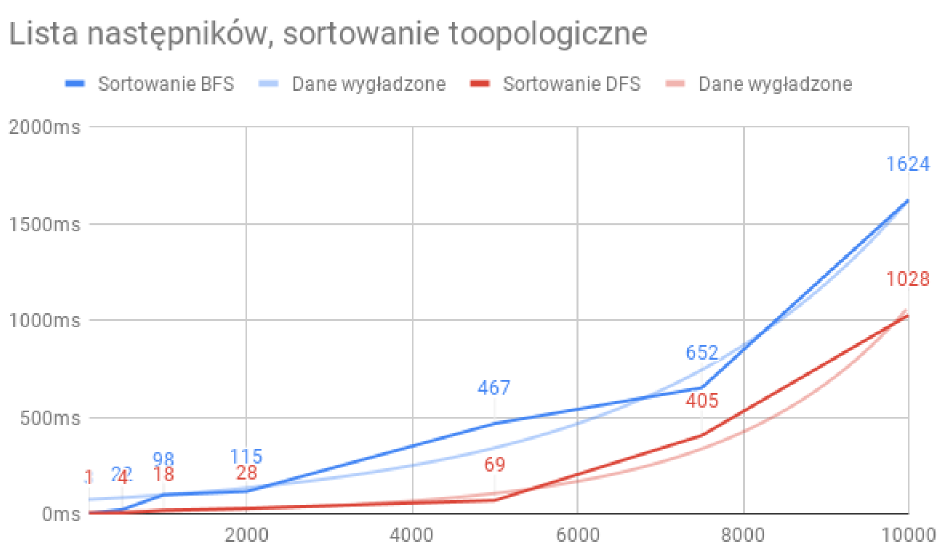
3.2.2 Znalezienie wszystkich następników

Cechuje się złożonością:

$$O(n) \tag{14}$$

która jest spowodowana potrzebą przejrzenia odpowiedniej listy, która w najgorszym przypadku może zawierać wszystkie wierzchołki

3.3 Sortowanie topologiczne



3.3.1 Sortowanie za pomocą algorytmu DFS

Wykonuje się ze złożonością

$$O(n^2) \tag{15}$$

powstaje one przez:

$$\text{ilośc operacji} \approx \text{ilość operacji znalezienia nastepników} * n \approx n * n \approx n^2 \tag{16}$$

3.3.2 Sortowanie za pomocą algorytmu BFS

Cechuje się złożonością:

$$\text{ilość operacji} \approx \text{ilość operacji szukania następników} * n \approx n * n \approx n^2 \tag{17}$$

Zastosowane oznaczenia

A dowolny wierzchołek w G, różny od B. 2, 4, 6, 8

B dowolny wierzchołek w G, różny od A. 2, 4, 6, 8

G skierowany graf acykliczny, w którym wierzchołki są etykietowane kolejnymi liczbami naturalnymi zaczynając od 0. 2, 4–6, 8

m ilość krawędzi w grafie. 8

n ilość wierzchołków w grafie. 2, 8