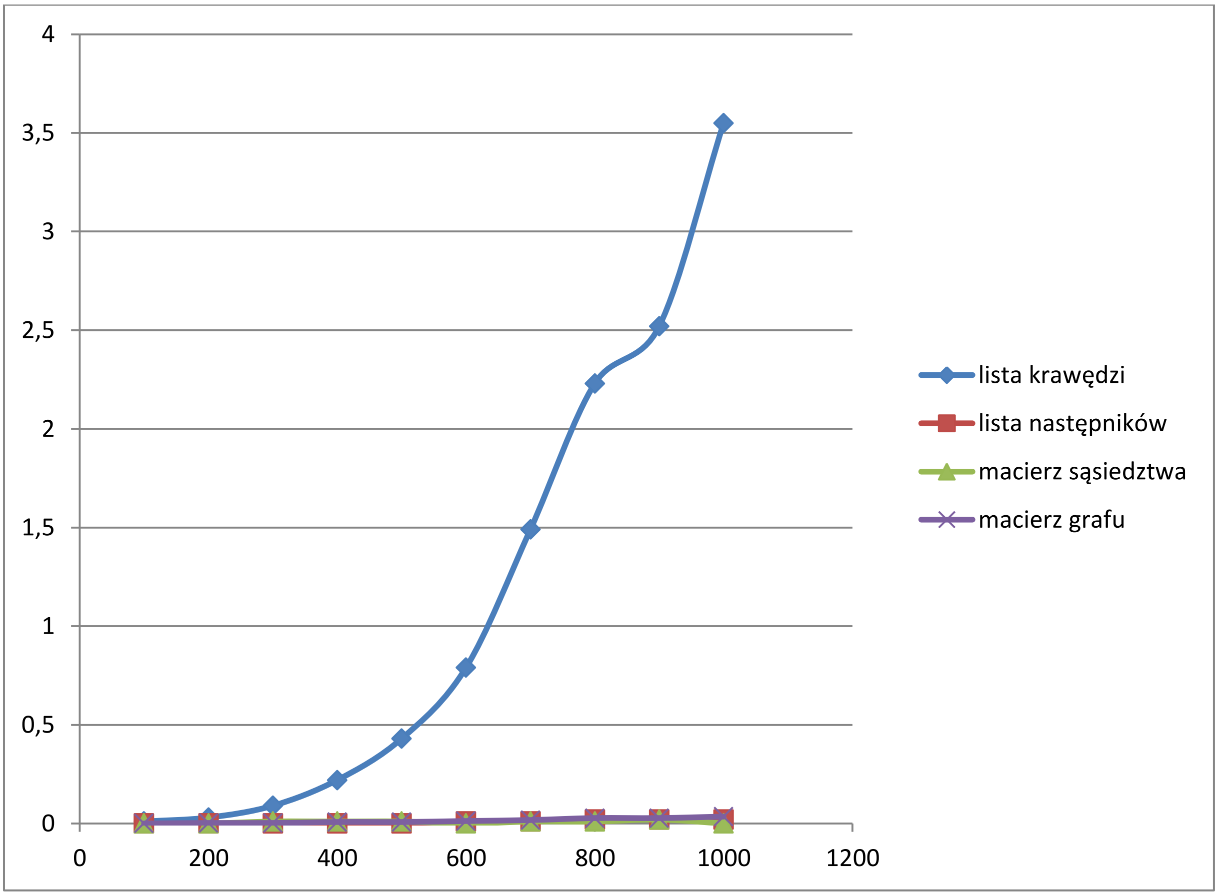
Sortowanie topologiczne – sprawozdanie

Przez sortowanie topologiczne rozumiemy takie ułożenie wierzchołków grafu, że jeżeli istnieje krawędź z x do y, to x będzie znajdował się przed y. W taki sposób posortować można tylko acykliczne grafy skierowane, ponieważ zawsze z pozostałych (jeszcze nieposortowanych) wierzchołków musi być taki, którego aktualny stopień wejściowy jest równy zero. Jeżeli w grafie istnieje cykl, to żaden z wierzchołków do niego należących nie będzie miał zerowego stopnia wejściowego.

Sortowanie topologiczne na podstawie algorytmu BFS



Sortowanie topologiczne z wykorzystaniem algorytmu przeszukiwania wszerz polega na utworzeniu na początku tablicy ze stopniami wejściowymi wszystkich wierzchołków. Te wierzchołki, których stopień wejściowy jest równy zero, popychane są do kolejki. Dopóki kolejka nie będzie pusta, zdejmujemy wierzchołek znajdujący się na początku i wszystkim jego następnikom zmniejszamy w tablicy stopień wejściowy o jeden. Jeżeli któryś z następników ma teraz stopień wejściowy równy zero, popychany jest do kolejki. Ostatecznie w kolejce musi znaleźć się każdy z n wierzchołków, więc pętla dopóki powtórzy się n razy. Złożoność algorytmu zależy jednak przede wszystkim od złożoności wyszukiwania następników, które następuje w owej pętli. Dostęp do następników jest zależny od reprezentacji grafu, dla której zaimplementujemy algorytm, dlatego złożoność sortowania topologicznego będzie zależna od tego, w jaki sposób przedstawiliśmy graf.

Reprezentacja grafu – lista następników

- Złożoność pamięciowa O(n+m)
- Złożoność znalezienia krawędzi:
  - pesymistyczna O(n) – przy założeniu, że nie jest to multigraf
  - średnia $O\left(\frac{m}{n}\right)$
- Złożoność znalezienia wszystkich następników:
  - pesymistyczna O(n) – przy założeniu, że nie jest to multigraf
  - średnia $O\left(\frac{m}{n}\right)$
- Złożoność sortowania topologicznego z wykorzystaniem BFS dla listy następników O(n+m)

Reprezentacja grafu za pomocą listy następników polega na utworzeniu tablicy n list, na których dla każdego z n wierzchołków pamiętane są jego następniki. Następników w grafie jest tyle, co krawędzi, czyli m. Złożoność pamięciowa takiej reprezentacji wynosi zatem O(n+m), ponieważ pamiętamy n numerów wierzchołków i m następników. Znalezienie intersującej nas krawędzi polega na przejściu listy następników wierzchołka, z którego ma wychodzić krawędź. Złożoność takiej operacji wynosi w pesymistycznym przypadku O(n), gdy z ów wierzchołka wychodzą krawędzie do każdego z pozostałych. Średnio jednak na każdej z list następników dla poszczególnych wierzchołków jest $\frac{m}{n}$ elementów. Złożoność znalezienia wszystkich następników jest taka sama, jak znalezienie jednego, ponieważ też polega przejściu przez listę następników wierzchołka.

Złożoność sortowania topologicznego wynosi O(n+m), ponieważ dla każdego z n wierzchołków w pętli dopóki kolejka nie jest pusta, przeglądamy tylko listę jego następników, zmniejszając ich stopień wejściowy. Łącznie następników na wszystkich listach jest m, stąd owa złożoność.

Reprezentacja grafu – macierz sąsiedztwa

- Złożoność pamięciowa O(nxn)
- Złożoność znalezienia krawędzi O(1) – czas stały
- Złożoność znalezienia wszystkich następników O(n)
- Złożoność sortowania topologicznego z wykorzystaniem BFS dla macierzy sąsiedztwa $O(n^2)$

Reprezentacja grafu za pomocą macierzy sąsiedztwa polega na utworzeniu tablicy dwuwymiarowej o rozmiarach nxn. Jeżeli istnieje krawędź z x do y, w macierzy na przecięciu numerów tych wierzchołków jest jeden, w przeciwnym przypadku zero. Daje to możliwość natychmiastowego sprawdzenia, czy istnieje krawędź. Wystarczy zobaczyć, czy w tablicy na przecięciu numerów wierzchołków jest jeden. Znalezienie wszystkich następników wierzchołka polega na sprawdzeniu całego wiersza tablicy. Tam gdzie w macierzy jest jeden, następnikiem jest indeks kolumny.

Złożoność sortowania topologicznego wynosi $O(n^2)$, ponieważ dla każdego z n wierzchołków wykonujemy operację szukania jego następników o złożoności O(n).

Reprezentacja grafu – tabela krawędzi

- Złożoność pamięciowa O(2xm)
- Złożoność znalezienia krawędzi O(m)
- Złożoność znalezienia wszystkich następników O(m)

- Złożoność sortowania topologicznego z wykorzystaniem BFS dla tabeli krawędzi O(n∗m)

Reprezentacja grafu za pomocą tabeli krawędzi polega na utworzeniu dwukolumnowej tablicy o m wierszach. W każdym kolejnym wierszu zapisana jest następna krawędź zgodnie z zasadą, iż jeżeli istnieje ona z x do y, to w pierwszej kolumnie zapisujemy x, a w drugiej y. Nietrudno sobie wyobrazić zatem, że, aby znaleźć krawędź, musimy po prostu przejść przez całą tablicę i sprawdzić, czy znajdziemy taką, której szukamy. By znaleźć wszystkie następniki wierzchołka również przechodzimy całą tablicę i jeżeli w pierwszej kolumnie mamy jego numer, to w drugiej kolumnie właśnie znaleźliśmy jeden z następników.

Złożoność sortowania topologicznego wynosi O(m∗n), ponieważ dla każdego z n wierzchołków wykonujemy operację wyszukiwania następników o złożoności O(m).

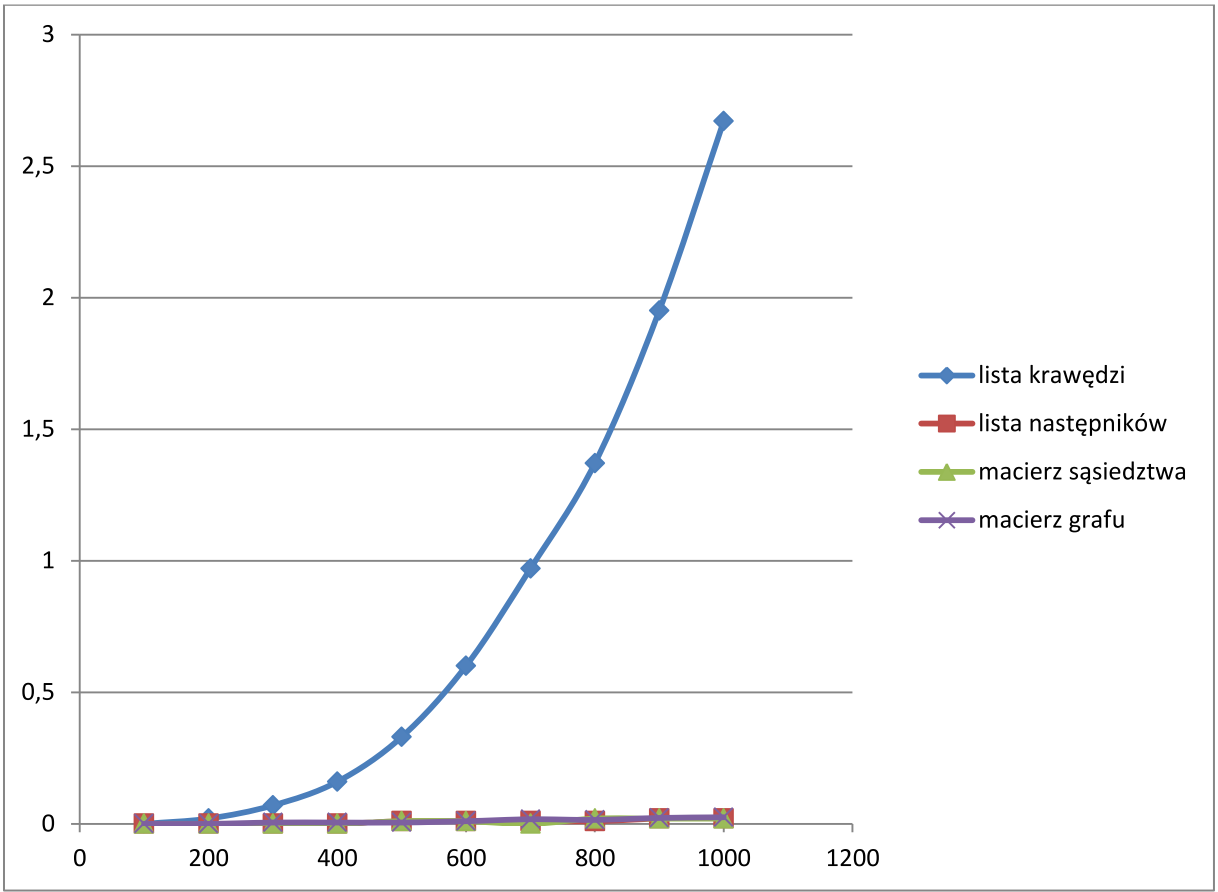
Reprezentacja grafu – macierz grafu

- Złożoność pamięciowa O(nx(n+3))
- Złożoność znalezienia krawędzi O(1) – czas stały
- Złożoność znalezienia wszystkich następników O(n)
- Złożoność sortowania topologicznego z wykorzystaniem BFS dla macierzy grafu O(n+m)

Reprezentacja grafu za pomocą macierzy grafu polega na utworzeniu tablicy dwuwymiarowej o rozmiarach nx(n+3). W pierwszych trzech kolumnach znajdują się kolejno pierwszy następnik, pierwszy poprzednik i pierwszy wierzchołek, który nie jest ani poprzednikiem, ani następnikiem wierzchołka o numerze takim, jak indeks wiersza. Następne kolumny uzupełniane są według zasady, że jeżeli wierzchołek jest następnikiem, to w kolumnie o swoim numerze zwiększonym o trzy ma zapisany ma numer kolejnego następnika. Ostatni następnik zaś w kolumnie o swoim numerze zwiększonym o trzy ma swój numer. Gdy wierzchołek jest poprzednikiem, to w kolumnie o swoim numerze zwiększonym o trzy ma numer następnego poprzednika przemnożony przez -1. Ostatni poprzednik zaś w kolumnie o swoim numerze zwiększonym o trzy ma swój numer również przemnożony przez -1. Brak incydencji wierzchołków zaznaczamy w taki sam sposób, lecz zamiast mnożyć przez -1 dodajemy n. Sprawdzenie, czy istnieje krawędź odbywa się w czasie stałym, wystarczy zobaczyć, czy w macierzy na przecięciu indeksów (jednego zwiększonego o trzy) znajduje się liczba z przedziału <1, n>. Znalezienie wszystkich następników może polegać na sprawdzeniu całego wiersza, w których kolumnach będziemy mieć wartości z przedziału <1, n>, wtedy złożoność wynosi O(n). Możemy jednak potraktować to, co mamy w kolumnie każdego kolejnego następnika jako adres kolejnego. Jest to swoista lista następników, po której możemy się w ten sposób przemieszczać. Złożoność takiej implementacji wyszukiwania następników to również O(n), ponieważ w pesymistycznym przypadku wszystkie liczby mogą być następnikami danego wierzchołka.

Dla każdego z wierzchołków w macierzy grafu mamy utworzoną swoistą listę następników, dlatego przy odpowiedniej implementacji wyszukiwania wszystkich następników, złożoność sortowania topologicznego będzie taka sama, jak dla reprezentacji grafu za pomocą listy następników.

Sortowanie topologiczne na podstawie algorytmu DFS



Sortowanie topologiczne z wykorzystaniem algorytmu przeszukiwania w głąb polega na wywołaniu funkcji, która dla nieodwiedzonych jeszcze wierzchołków wywoła funkcję rekurencyjną. Funkcja rekurencyjna oznacza wierzchołek jako odwiedzony i wywołuje samą siebie dla wszystkich następników oraz po tym wywołaniu odkłada dany wierzchołek na stos wynikowy. Takie odkładanie na stos (po uprzednim wywołaniu funkcji dla wszystkich następników) zapewnia, że wszystkie następniki odkładanego wierzchołka już się na nim znajdują. Funkcja rekurencyjna wywoływana jest raz dla każdego wierzchołka, bo warunkiem jej wywołania jest to, że nie został on jeszcze odwiedzony. Mamy więc podobną sytuację, co w przypadku sortowania topologicznego przy pomocy algorytmu BFS, n razy powtórzymy operację wyszukiwania następników. Złożoność tego algorytmu zatem również zależy od reprezentacji grafu, dla której zaimplementowaliśmy algorytm i jest taka sama jak w przypadku sortowania topologicznego przy pomocy algorytmu przeszukiwania grafu wszerz.

- Złożoność sortowania topologicznego z wykorzystaniem DFS dla tabeli krawędzi $O(n*m)$
- Złożoność sortowania topologicznego z wykorzystaniem DFS dla macierzy sąsiedztwa $O(n^2)$
- Złożoność sortowania topologicznego z wykorzystaniem DFS dla listy następników $O(n+m)$
- Złożoność sortowania topologicznego z wykorzystaniem DFS dla macierzy grafu $O(n+m)$