Jakob Wolitzki, grupa I2, nr albumu: 136830

Ćwiczenie nr 3.

Temat: Algorytmy grafowe

Wymagania:

1. Definicja grafu skierowanego i nieskierowanego.
2. Znajomość następujących reprezentacji grafu w maszynie cyfrowej:
   - macierz sąsiedztwa
   - macierz incydencji
   - lista krawędzi
   - lista incydencji

oraz znajomość złożoności pamięciowej dla tych reprezentacji i złożoności czasowej uzyskiwania informacji o wierzchołkach i krawędziach grafu dla każdej z tych reprezentacji.

3. Znajomość procedury przeglądania grafu w głąb i wszerz.
4. Znajomość algorytmu sortowania topologicznego i jego złożoności obliczeniowej.

Przebieg ćwiczenia:

1. Dla losowo wygenerowanego grafu wejściowego nieskierowanego (graf wejściowy jest wczytywany z pliku tekstowego w taki sposób, aby możliwe było wpisanie dowolnego grafu) zbadaj czas pozyskania informacji o istnieniu krawędzi pomiędzy parą wylosowanych wierzchołków, dla każdej z wymienionych powyżej reprezentacji grafu. Badania przeprowadzić dla wszystkich elementów wybierając losowo poszczególne wierzchołki. Obliczyć średni czas poszukiwania, sumując czasy szukania poszczególnych elementów i dzieląc tę sumę przez liczbę elementów (liczba wierzchołków w grafie - *n*, nasycenie grafu krawędziami 0.6). Wykres *t=f(n)*, dla każdej reprezentacji; na jednym wykresie, dla każdego poszukiwania.

2. Dla losowo wygenerowanego grafu DAG (n - wierzchołków i współczynniku wypełnienia krawędziami 0.3), zastosować procedurę sortowania topologicznego Wykres *t=f(n)*- czasu tworzenia posortowanego ciągu. Uzasadnij wybór reprezentacji grafu. Podaj zalety i wady wybranej reprezentacji w porównaniu z pozostałymi.

3. Sformułować wnioski odnośnie efektywności każdej z reprezentacji grafu oraz problemu sortowania topologicznego.

Czas realizacji: 2 tygodnie

Zad 1.

Zacznę od opisania poszczególnych reprezentacji grafów.

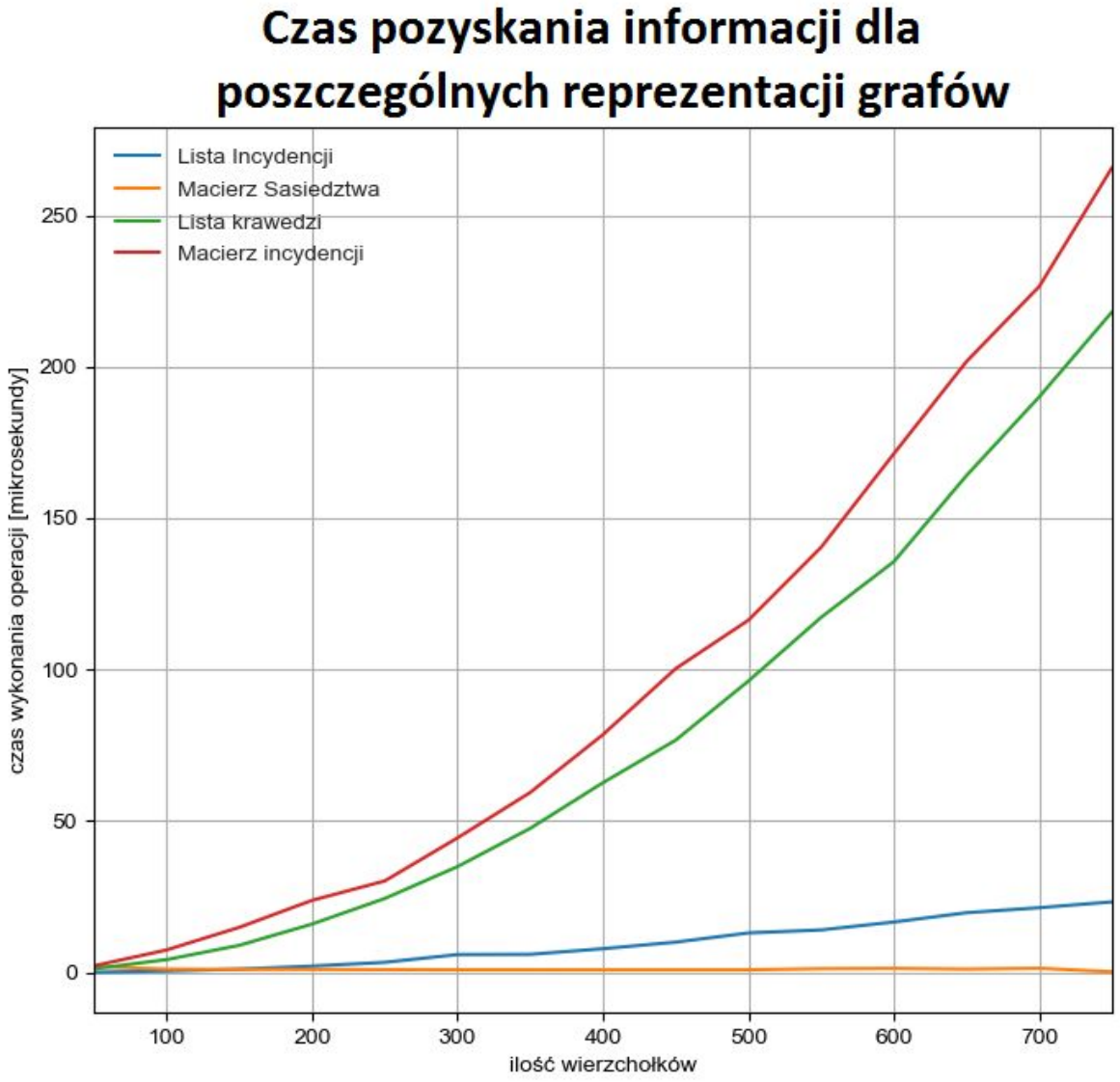
Macierz sąsiedztwa, ta reprezentacja grafu bazuje na macierzy kwadratowej n x n, gdzie n jest ilością wierzchołków grafu. Pokazuje ona połączenia wierzchołków krawędziami. Wiersze macierzy odwzorowują początki krawędzi, a kolumny ich końce. W naszym przypadku nie ma to jednak znaczenia, gdyż grafy do tego zadania są nieskierowane. Dla grafów nieskierowanych, macierz sąsiedztwa jest symetryczna względem głównej przekątnej.

Macierz incydencji, w tej reprezentacji grafu, macierz jest o wymiarze n x m, gdzie n oznacza liczbę wierzchołków, a m oznacza liczbę krawędzi. Wiersze, odpowiadają wierzchołkom grafu, a kolumny jej krawędzią. W każdej kolumnie muszą być dwie jedynki, które odpowiadają połączeniu pomiędzy wierzchołkami. Gdyby graf był nieskierowany, to -1 oznaczałoby, że z wierzchołka wychodzi krawędź, a 1 ,że do niego wchodzi.

Lista krawędzi, ta reprezentacja grafu odpowiada tablicy, która przechowuje structy zawierające dwie zmienne, początek wierzchołka i jego koniec.

Lista incydencji, w tej reprezentacji grafu, użyta jest tablica n elementowa, gdzie n oznacza liczbę wierzchołków. Każdym elementem tablicy jest lista. W każdej liście, jej początkiem jest wierzchołek startowy. Są tam również przechowywane numery wierzchołków, które odpowiadają połączeniom pomiędzy wierzchołkiem początkowym a jego sąsiadami. Warto wspomnieć, że w przypadku grafów nieskierowanych, ich listy są dłuższe, gdyż muszą pokazywać krawędzie w obu kierunkach.

Wykresy, które opracowałem na podstawie wyników z przeprowadzonych pomiarów nie są idealnym odzwierciedleniem funkcji opisujących złożoności obliczeniowe danych operacji. Występują w nich odchylenia, które spowodowane są np. niedokładnie jednorodnym rozkładem danych losowych, spowodowanym działaniem wbudowanych funkcji losujących języka C++, niepewnościami pomiarowymi związanymi z pomiarem czasów przez funkcje wbudowane języka C++, oraz obliczeniami z ograniczoną dokładnością pomiarową wykonanymi przez komputer.



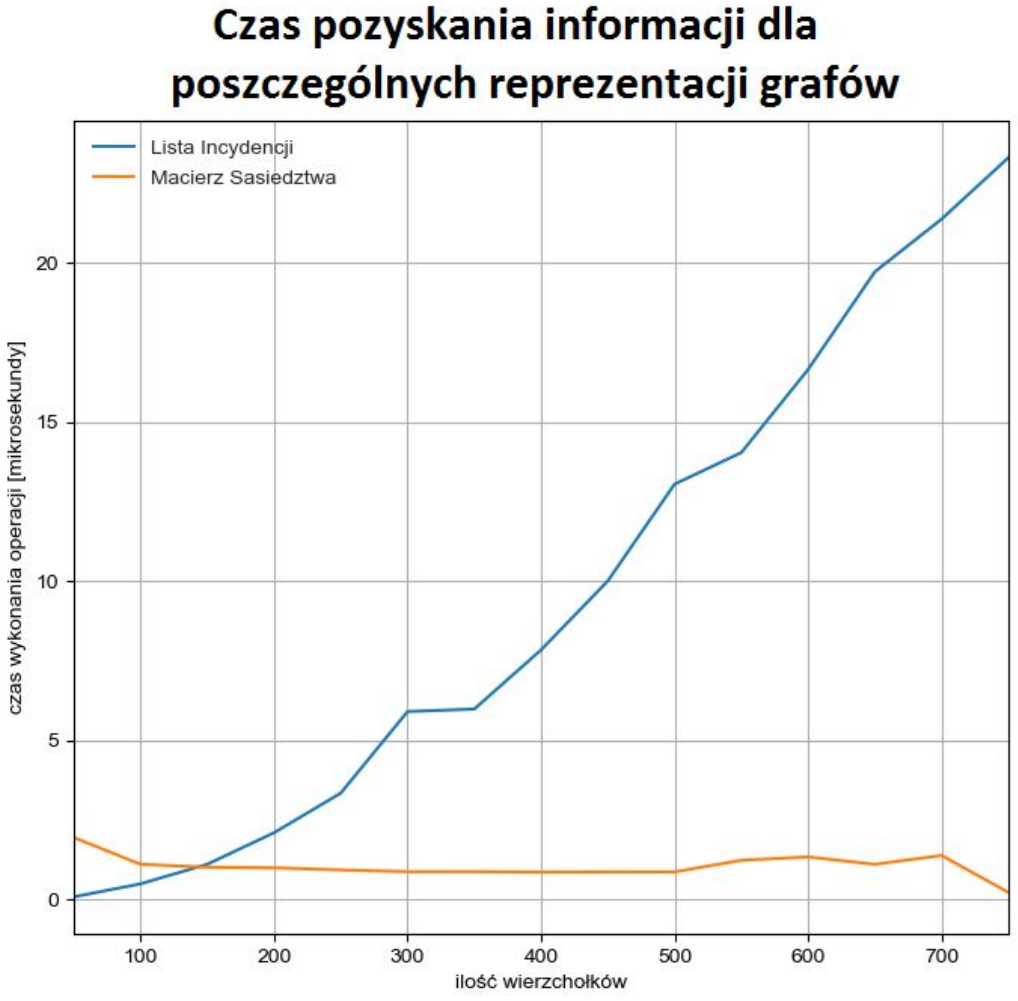
Najszybsza okazała się być macierz sąsiedztwa. Jej złożoność obliczeniowa pozyskiwania informacji jest stała. Czas potrzebny na wykonanie tej operacji jest stały, nie zależy od ilości wierzchołków. Jej wadą jest wysoka złożoność pamięciowa, która wynosi $O(n^2)$, gdyż składa się ona z macierzy n x n, gdzie n to ilość wierzchołków. Dlatego nie zaleca się jej stosowania do grafów rzadkich.

Kolejną bardzo szybką reprezentacją grafu są listy incydencji. Jej złożoność jest rzędu O(n). Wynika on z faktu, iż aby zbadać czy dane wierzchołki są połączone krawędzią, w najgorszym przypadku należy przejrzeć całą listę sąsiadów danego wierzchołka.
Czas potrzebny do przeszukania większej ilości krawędzi wynosi zatem O(n*m), gdzie m to liczba krawędzi. Złożoność pamięciowa list incydencji wynosi O(n+m). Przy grafie nieskierowanym, suma wszystkich list wynosi m, jednak w naszym przypadku wynosi 2*m. Jest dobrą reprezentacją dla przedstawiania grafów rzadkich.

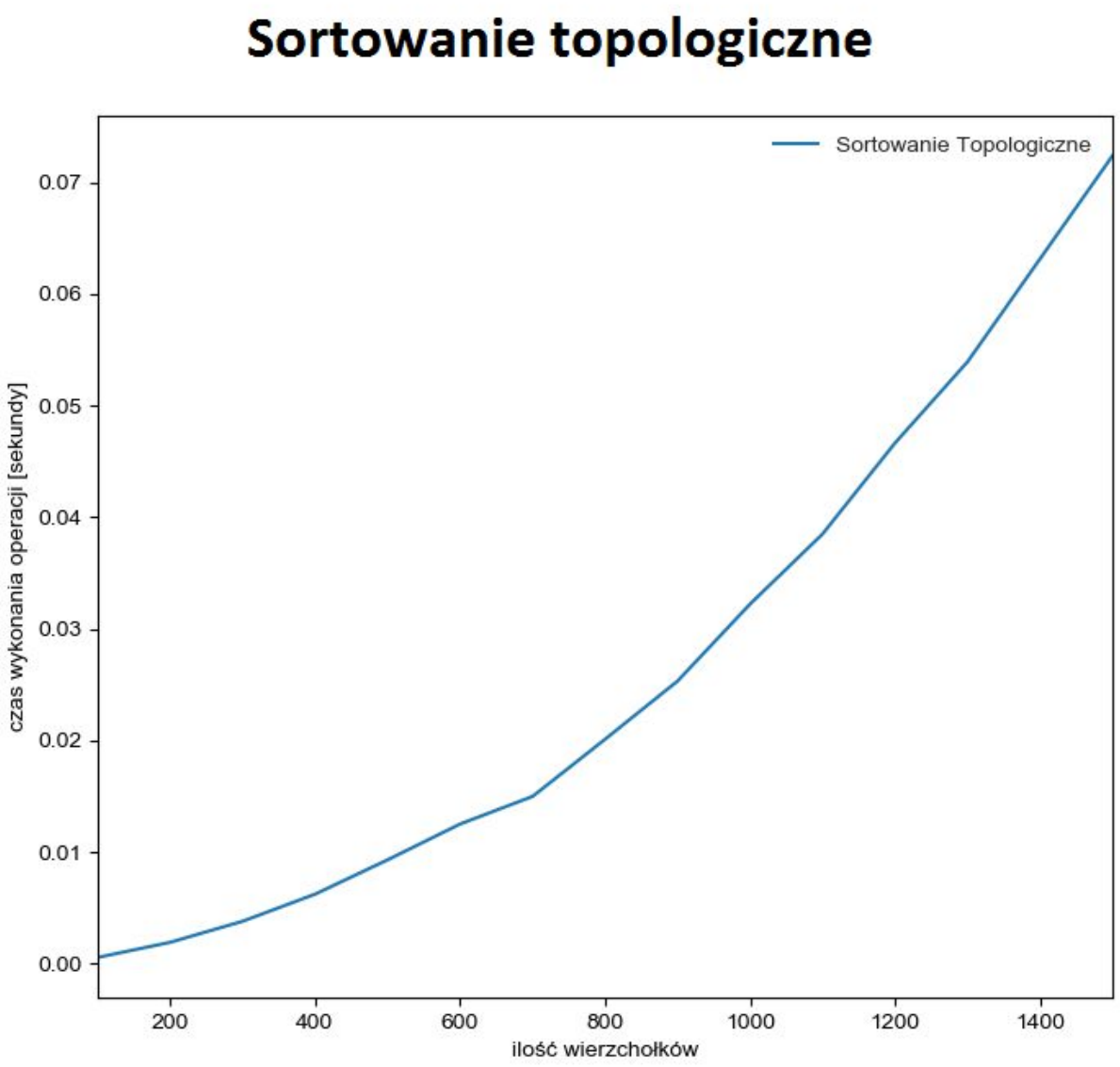
Złożoność obliczeniowa sprawdzania istnienia pojedynczej krawędzi za pomocą listy krawędzi wynosi O(m), gdzie m to ilość krawędzi. Wynika to z faktu, iż w najgorszym przypadku algorytm musi przejść przez całą listę krawędzi (gdy krawędź jest na końcu lub jej nie ma). Zatem gdy wybieramy losowo poszczególne pary wierzchołków, czas potrzebny na szukanie ich jest rzędu $O(m^2)$. Warto wspomnieć o zalecie listy krawędzi, jest nią jej złożoność pamięciowa, która wynosi O(m), gdzie m to liczba krawędzi.

Podobnie szybka do listy krawędzi okazała się być macierz incydencji. Złożoność obliczeniowa operacji szukania informacji o istnieniu krawędzi jest rzędu O(m), gdzie m to liczba krawędzi. Jest tak, gdyż w najgorszym przypadku należy przejrzeć cały wiersz o długości m, gdy szukana krawędź znajduje się na końcu lub jej nie ma. Czas potrzebny na uzyskanie informacji o istnieniu wszystkich krawędzi wynosi zatem $O(m^2)$, gdzie m to liczba krawędzi. Złożoność pamięciowa macierzy incydencji jest rzędu O(n*m), dlatego nie powinno się jej używać do reprezentowania grafów gęstych.

Poniżej umieszczam również dokładniejszy wykres pokazujący różnice pomiędzy reprezentacjami list incydencji i macierzy sąsiedztwa.



Zad 2.



Sortowanie topologiczne grafu polega na posortowaniu wierzchołków w taki sposób, aby każdy wierzchołek posiadający sąsiadów znalazł się przed nimi, każdy wierzchołek musi poprzedzać te, do których prowadzą wychodzące od niego krawędzie. Aby można było posortować graf, musi on być skierowany, i acykliczny (DAG).

Przy zastosowaniu procedury sortowania topologicznego, wykorzystuję algorytm przeszukiwania grafu w głąb (DFS). Sortowanie ma złożoność obliczeniową rzędu O(n+m), gdzie n to liczba wierzchołków, a m to liczba krawędzi. Czas wstawiania każdego z wierzchołków do tablicy wynikowej jest stały. Algorytm BFS jest wykonywany dla każdego nieodwiedzonego wierzchołka i dla każdego nieodwiedzonego wierzchołka połączonego krawędzią (rekurencyjnie).

W moim algorytmie sortowania topologicznego zastosowałem Listy incydencji, gdyż bardzo łatwo jesteśmy w stanie odwołać się do kolejnego wierzchołka, połączonego krawędzią. Jest to rozwiązanie proste, łatwe w implementacji oraz zapewnia szybki dostęp do informacji o istnieniu krawędzi (O(n)). Zaletą tej implementacji jest również małe wykorzystanie pamięci, w porównaniu do stworzenia np. macierzy sąsiedztwa.
W przypadku użycia macierzy incydencji czy listy krawędzi, czas użyty na znalezienie sąsiadów pojedynczego wierzchołka byłby rzędu O(m), co bardzo spowolniłoby czas działania sortowania topologicznego.

Zad 3.
Wnioski, podsumowanie.

Uzyskiwanie informacji o krawędzi w grafie jest najszybsze dla macierzy sąsiedztwa. Operacja ta jest wykonywana w czasie stałym. Wadą tej reprezentacji jest wysokie zużycie pamięci, złożoność $O(n^2)$. Jeżeli jednak zależy nam na szybkim dostępie do informacji, a nie koniecznie na pamięci to reprezentacja ta jest najlepsza, dodatkowo bardzo prosta w implementacji.

Listy incydencji są drugim pod względem szybkości rodzajem reprezentacji grafów. Złożoność obliczeniowa znalezienia pojedynczej krawędzi wynosi O(n), gdzie n jest liczbą wierzchołków. Złożoność pamięciowa jest rzędu O(n+m), gdzie m to liczba krawędzi. Dlatego porównując tą reprezentację z macierzą sąsiedztwa, korzystniej wypada reprezentacja listowa (dla grafów rzadkich).

Macierz incydencji oraz lista krawędzi wypadły bardzo podobnie na wykresie porównującym szybkości reprezentacji grafów. Lista krawędzi okazała się być niewiele szybsza od macierzy incydencji. Jej złożoność pamięciowa jest bardzo mała, wynosi O(m), co jest ogromną zaletą. Złożoność pamięciowa macierzy incydencji wynosi O(m*n). Złożoność uzyskiwania informacji o pojedynczym połączeniu wierzchołków jest w obu reprezentacjach taka sama i wynosi O(m). Macierz incydencji jest zatem bardzo nieefektywną reprezentacją zarówno jeżeli bierzemy pod uwagę złożoność pamięciową jak i obliczeniową, pozyskiwania informacji.

Sortowanie topologiczne ma złożoność obliczeniową rzędu O(n+m), gdzie n jest liczbą wierzchołków, a m krawędzi. Duże znaczenie na szybkość sortowania ma współczynnik zagęszczenia grafu, w naszym przypadku wynosi on 0,3 czyli graf jest rzadki. Jednak gdyby graf był gęsty, to całe sortowanie trwałoby znacznie dłużej. Również szybkość działania sortowania topologicznego zależy od reprezentacji grafu. Jeżeli bierzemy pod uwagę oszczędność pamięci oraz szybkość wykonywania algorytmu to najlepszą implementacją jest użycie list incydencji.