Laboratorium Algorytmów i Struktur Danych			
Kierunek:Informatyka	Specjalność: -	Rok: I	Grupa: I2
Algorytmy grafowe *(23.04.2017r.)*			Nr 3
Kinga Karasiewicz (*132247*)			
Uwagi:			Ocena:

1. Cel ćwiczenia

Celem laboratoriów jest zbadanie właściwości i skuteczności wybranych reprezentacji grafów. Macierzy sąsiedztwa (MN), listy następników (NA) oraz listy łuków (AA). Innym celem jest także zastosowanie sortowania topologicznego składającego się z obliczenia dla każdego wierzchołka etykiet czasowych rozpoczęcia i zakończenia analizy (dla wybranej reprezentacji grafu), a także sprawdzanie acykliczności poprzez zliczanie łuków powrotnych dla wszystkich reprezentacji grafowych.

2. Podstawy teoretyczne

Graf - graf to zbiór wierzchołków, które mogą być połączone krawędziami w taki sposób, że każda krawędź kończy się i zaczyna w którymś z wierzchołków. Wierzchołki grafu mogą być numerowane, natomiast krawędzie mogą obrazować relacje między takimi obiektami. Wierzchołki należące do krawędzi nazywane są jej końcami. Krawędzie mogą mieć wyznaczony kierunek, a graf zawierający takie krawędzie nazywany jest grafem skierowanym.

Macierz sąsiedztwa - macierz kwadratowa, w której *aij* oznacza liczbę krawędzi pomiędzy wierzchołkami i. W przypadku grafów prostych macierz sąsiedztwa jest macierzą zerojedynkową z zerami na głównej przekątnej. Dla grafów nieskierowanych macierz sąsiedztwa jest z definicji symetryczna.

- złożoność pamięciowa: $O(n^2)$
- przejrzenie wszystkich krawędzi: $O(n^2)$
- przejrzenie następników/poprzedników danego wierzchołka: O(n)
- sprawdzenie istnienia jednej krawędzi: O(1)

Lista następników - dla każdego wierzchołka zapamiętywana jest lista następników wierzchołków. W implementacji tej metody stosuje się listy jednokierunkowe oraz jednowymiarową tablicę wskaźników.

- złożoność pamięciowa: O(n+m)
- przejrzenie wszystkich krawędzi: O(n+m)
- przejrzenie następników danego wierzchołka: maksymalnie O(n) (może być mniejsza)
- przejrzenie poprzedników danego wierzchołka: O(n+m)
- sprawdzenie istnienia jednej krawędzi: maksymalnie O(n), średnio jest mniejsza

Lista poprzedników - dla każdego wierzchołka zapamiętywana jest lista poprzedników wierzchołków. W implementacji tej metody stosuje się listy jednokierunkowe oraz jednowymiarową tablicę wskaźników.

- złożoność pamięciowa: O(n+m)
- przejrzenie wszystkich krawędzi: O(n+m)
- przejrzenie następników danego wierzchołka: maksymalnie O(n+m)
- przejrzenie poprzedników danego wierzchołka: O(n) (może być mniejsza)
- sprawdzenie istnienia jednej krawędzi: maksymalnie O(n), średnio jest mniejsza

Lista łuków - lista pary wierzchołków (początki i końce krawędzi) bez żadnej specjalnej kolejności .

- złożoność pamięciowa: O(m)
- przejrzenie wszystkich krawędzi: O(m)
- przejrzenie następników/poprzedników danego wierzchołka: O(m)
- sprawdzenie istnienia jednej krawędzi: O(m)

Macierz incydencji - składa się z V wierszy i E kolumn. Na "skrzyżowaniu" wierzchołka z krawędzią jest -1 gdy krawędź wychodzi z wierzchołka, +1 - krawędź wchodzi, 2 - pętla, 0 - brak incydencji.

- złożoność pamięciowa: O(n^2)
- przejrzenie wszystkich krawędzi: O(n^2)
- przejrzenie następników/poprzedników danego wierzchołka: O(n)
- sprawdzenie istnienia jednej krawędzi: O(n)

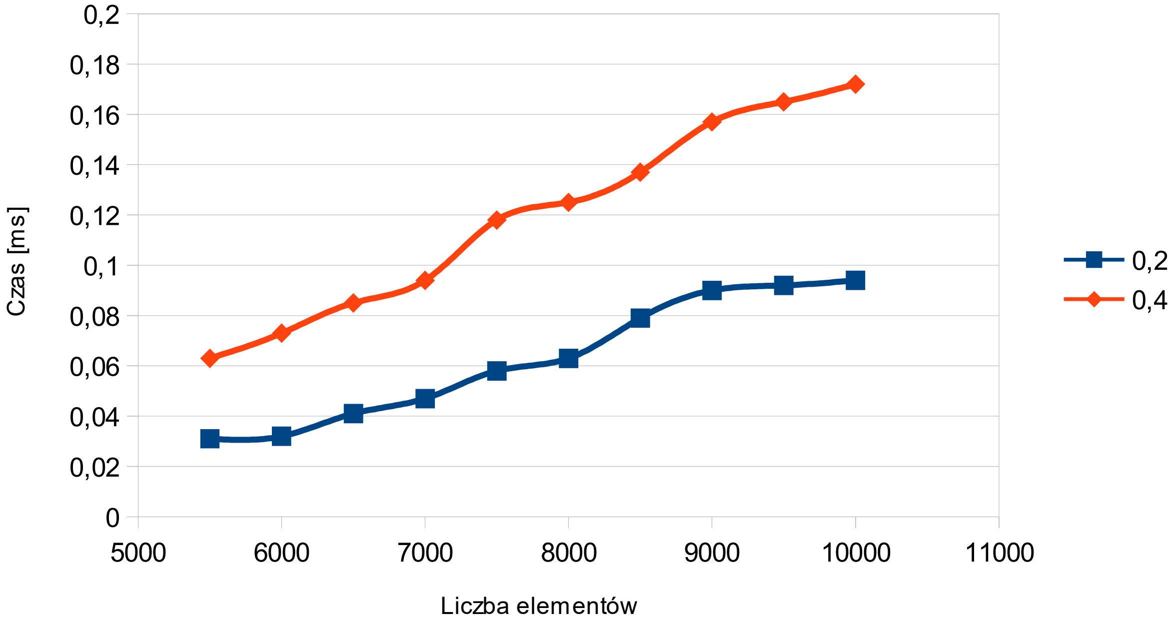
Sortowanie topologiczne - liniowe uporządkowanie wierzchołków, w którym jeśli istnieje krawędź skierowana prowadząca od wierzchołka do, to znajdzie się przed wierzchołkiem. Innymi słowy, każdy wierzchołek poprzedza wszystkie te wierzchołki, do których prowadzą wychodzące od niego krawędzie.
Wierzchołki w każdym grafie acyklicznym skierowanym można posortować topologicznie na jeden lub więcej sposobów.

3.1 Ćwiczenie pierwsze

Ćwiczenie polegało na badaniu zależności czasu trwania obliczania etykiet od liczby wierzchołków N w zależności od gęstości d.

Rozmiar (N)	d = 0,2	d = 0,4
5500	0,031	0,063
6000	0,032	0,073
6500	0,041	0,085
7000	0,047	0,094
7500	0,058	0,118
8000	0,063	0,125
8500	0,079	0,137
9000	0,090	0,157
9500	0,092	0,165
10000	0,094	0,172

Tabela 1: Tabela przedstawia czas trwania etapu obliczania etykiet w zależności od rozmiaru(N) i gęstości (d)



Ilustracja 1: Wykres przedstawia czas trwania etapu obliczania etykiet w zależności od rozmiaru(N) i gęstości (d)

3.2 Wnioski

Algorytm sortowania topologicznego opiera się na algorytmie przeszukiwania grafu w głąb. Jego drobna modyfikacja poprzez zapisywanie etykiet czasowych rozpoczęcia i zakończenia analizy każdego wierzchołka pozwala sprawdzić, czy graf jest acykliczny.

Efektywność tego algorytmu zależy od reprezentacji grafu. Najczęściej wykonywaną tam operacją jest wyznaczanie zbioru następników, dlatego też najlepszą do tego strukturą jest lista następników (NA), posiada ona bardzo szybką informację odnośnie następników bieżącego wierzchołka. Pesymistycznym wariantem dla tego algorytmu byłaby lista poprzedników, gdyż najdłużej zajmuje tam wyznaczenie listy następników.

Algorytm DFS przeszukuje wszystkie wierzchołki grafu i dla każdego z nich sprawdza ich następniki, zatem jego złożoność wynosi O(n+m). Im gęstszy graf, tym więcej łuków i tym dłuższy czas wykonywania algorytmu.

4.1 Ćwiczenie drugie

Ćwiczenie polegało na zbadaniu liczby łuków powrotnych w zależności od liczby wierzchołków N i gęstości d.

Rozmiar	d = 0,2	d = 0,4
5500	3025180	6048876
6000	3599685	7199766
6500	4222695	8449300
7000	4896324	9799317
7500	5623705	11246569
8000	6402761	12797695
8500	7223592	14448234
9000	8099846	16201465
9500	9027223	18049693
10000	10001006	19994304

Tabela 2: Tabela przedstawia liczę łuków powrotnych w zależności od rozmiaru(N) i gęstości (d)

4.2 Wnioski

Procedura zliczania łuków powrotnych analizuje wszystkie łuki i sprawdza czy dany łuk (U, V) spełnia warunek d[v] < d[u] < f[u] < f[v], gdzie d oznacza rozpoczęcie, a f zakończenie analizy danego wierzchołka. Jeżeli, żaden wierzchołek nie spełnia tej zależności to graf jest acykliczny. W naszym wypadku wszystkie analizowane grafy okazały się cykliczne.

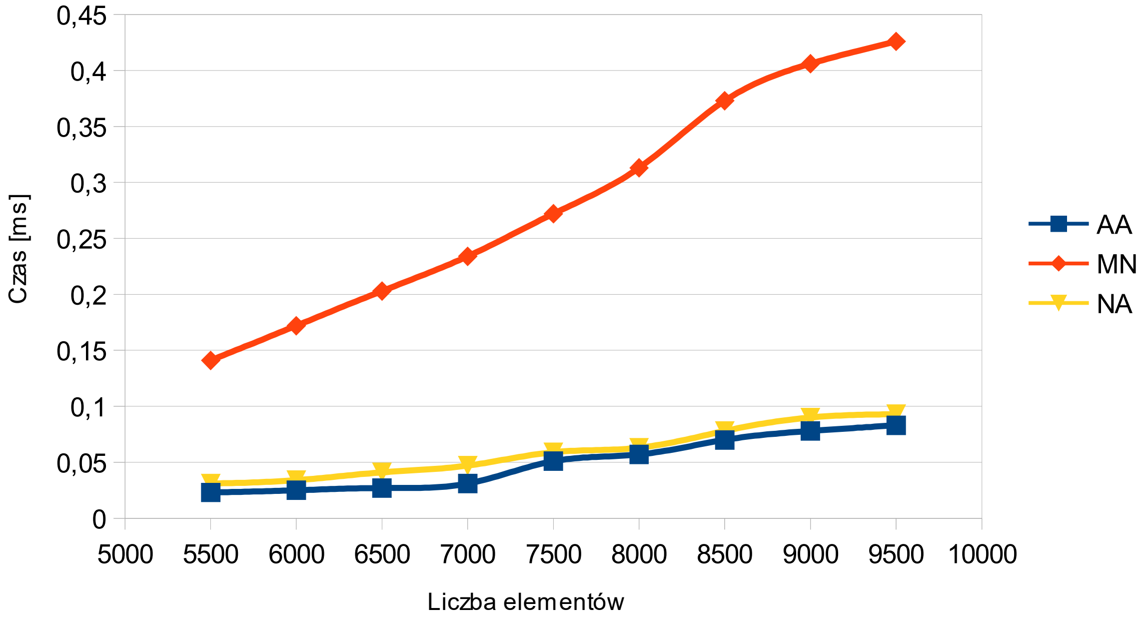
Sortować topologicznie możemy wtedy, gdy dany graf będzie acyklicznym grafem skierowanym. Czyli wtedy, gdy tworzymy listę wierzchołków grafu, to każdy wierzchołek posiadający sąsiadów znajduje się na liście przed nimi.

5.1 Ćwiczenie trzecie

Ćwiczenie polegało na zbadaniu czasu trwania etapu zliczania liczby łuków powrotnych od liczby wierzchołków N i gęstości d dla trzech reprezentacji grafu.

N	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
AA	0,023	0,025	0,027	0,031	0,051	0,057	0,070	0,078	0,083	0,094
MN	0,141	0,172	0,203	0,234	0,272	0,313	0,373	0,406	0,043	0,453
NA	0,031	0,034	0,041	0,047	0,059	0,063	0,078	0,090	0,093	0,094

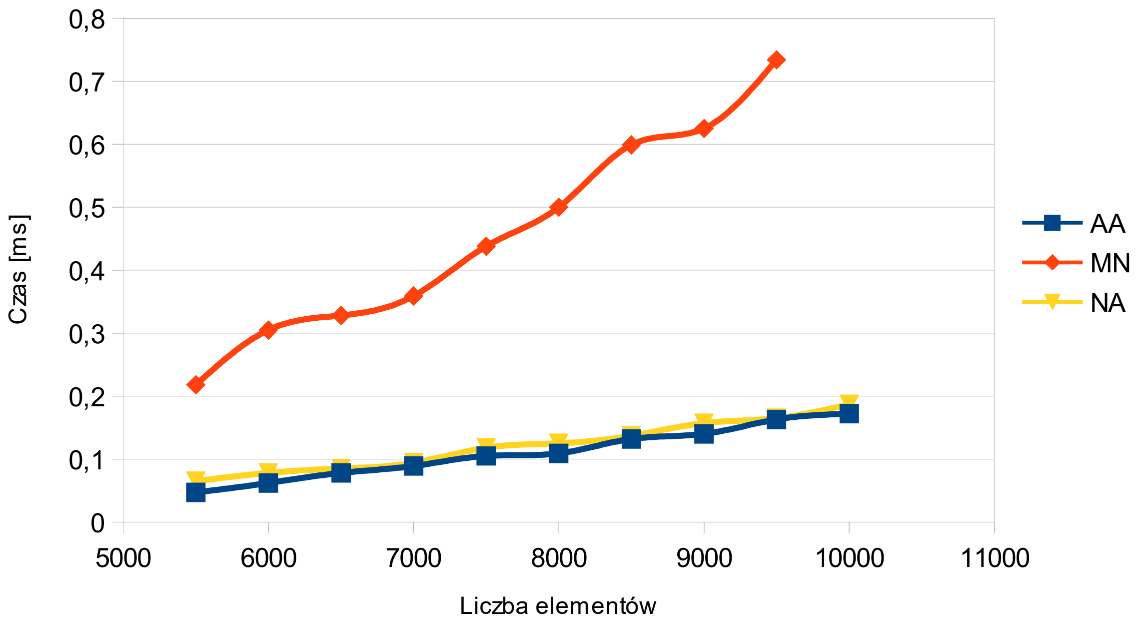
Tabela 3: Tabela przedstawia czas zliczania łuków powrotnych od rozmiaru(N) dla gęstości d=0,2



Ilustracja 2: Wykres przedstawia czas zliczania łuków powrotnych od rozmiaru(N) dla gęstości d=0,2

N	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
AA	0,047	0,062	0,078	0,089	0,105	0,115	0,132	0,140	0,163	0,172
MN	0,218	0,305	0,328	0,359	0,438	0,500	0,599	0,625	0,734	0,750
NA	0,065	0,078	0,085	0,094	0,118	0,125	0,137	0,157	0,165	0,187

Tabela 4: Tabela przedstawia czas zliczania łuków powrotnych od rozmiaru(N) dla gęstości d=0,4



Ilustracja 3: Wykres przedstawia czas zliczania łuków powrotnych od rozmiaru(N) dla gęstości d=0,4

5.2 Wnioski

Najkorzystniejszą reprezentacją grafu dla tej podprocedury jest lista łuków, dla której zależność wyznaczania łuków jest optymalna i wynosi O(m). Najgorszą strukturą jest macierz sąsiedztwa, gdzie wymagane jest sprawdzenie wszystkich łuków (także tych nieistniejących). Przy większej gęstości liczba łuków powrotnych zwiększa się. Jednocześnie tam gdzie zwiększa się gęstość grafu mocno zwiększa się czas przy powolniejszych reprezentacjach grafu jak macierz sąsiedztwa. Im więcej jest połączeń pomiędzy wierzchołkami (czyli większa gęstość), tym więcej należy ich sprawdzić i zliczyć.

6. Wnioski ogólne

Krótkie porównanie wymienionych reprezentacji grafu.

Najgorszą reprezentacją jest macierz incydencji. Nie wyróżnia się ona niczym szczególnym pośród innych reprezentacji grafów. Ma ogromną złożoność pamięciową $O(n^2)$, przejrzenie wszystkich krawędzi, teście krawędzi i przejrzeniu wszystkich następników i poprzedników wierzchołka wynosi O(n). Ta reprezentacja nie ma raczej zalet, jej jedną może być dosyć prosta implementacja.

Macierz sąsiedztwa jest już lepszą formą reprezentacji od macierzy incydencji. Ponieważ jest to macierz zero jedynkowa można zastosować tam bity i jej złożoność pamięciowa znacznie się zmniejsza. Największą jej zaletą jest test łuku, gdzie złożoność obliczeniowa wynosi O(1). W każdym innym przypadku ta złożoność wzrasta, gdyż przeglądając macierz, przeglądamy także nieistniejące łuki (tam, gdzie występuje 0). Ta macierz jest przydatna właśnie dla testu łuku.

Lista następników posiada mniejszą złożoność pamięciową O(n+m), gdyż nie zawiera nieistniejących łuków jak w poprzednich macierzach. Najlepszy czas osiąga dla przejrzenia listy wszystkich następników O(n)(jednak w praktyce jest mniejszy), lecz najdłuższy czas osiąga dla listy przeglądanie listy poprzedników O(m+n). Dlatego jej najlepszą cechą jest łatwa generacja listy następników.

Lista poprzedników jest podobna do listy następników, jednak w tym przypadku osiąga najlepszy czas do wygenerowania listy poprzedników, a najgorszy dla następników.

Lista łuków jest przydatna przy sortowaniu topologicznym i sprawdzeniu liczby łuków powrotnych. Jej złożoność pamięciowa wynosi O(m), dla reszty testów i generowania list złożoność obliczeniowa wynosi O(m).

7. Bibliografia:

- Wiadomości podane na ćwiczeniach z Algorytmów i Struktur Danych
- Informacje ze strony http://www.algorytm.org/klasyczne/grafy-i-ich-reprezentacje.html (dostęp z dnia 24.04.2017r.)
- Informacje ze strony http://www.cs.put.poznan.pl/arybarczyk/GrafReprezentacje.htm (dostęp z dnia 24.04.2017r.)
- Informacje ze strony http://eduinf.waw.pl/inf/alg/001_search/0122.php (dostęp z dnia 24.04.2017r.)