

# Algorytmy grafowe

Maciej Kaszkowiak, 151856

## 1 METODOLOGIA POMIARU WYDAJNOŚCI ALGORYTMÓW

---

Zmierzyłem czas wykonywania wybranych operacji dla następujących struktur danych:

- macierz sąsiedztwa
- macierz grafu

Zmierzone operacje dla powyższych struktur obejmowały:

- sortowanie topologiczne algorytmem Kahna
- sortowanie topologiczne z wykorzystaniem DFS

Algorytmy zostały zaimplementowane w języku Python 3.8.10. Testy zostały uruchomione pod systemem Windows 10 21H1.

Dane wejściowe zostały wygenerowane w następujący sposób:

- Graf acykliczny o współczynniku nasycenia 50%, utworzony poprzez rozpatrywanie krawędzi skierowanych od wierzchołka  $N$  do wierzchołka  $N+1$ ,  $N+2$ , ... oraz dodawanie co drugiej możliwości.

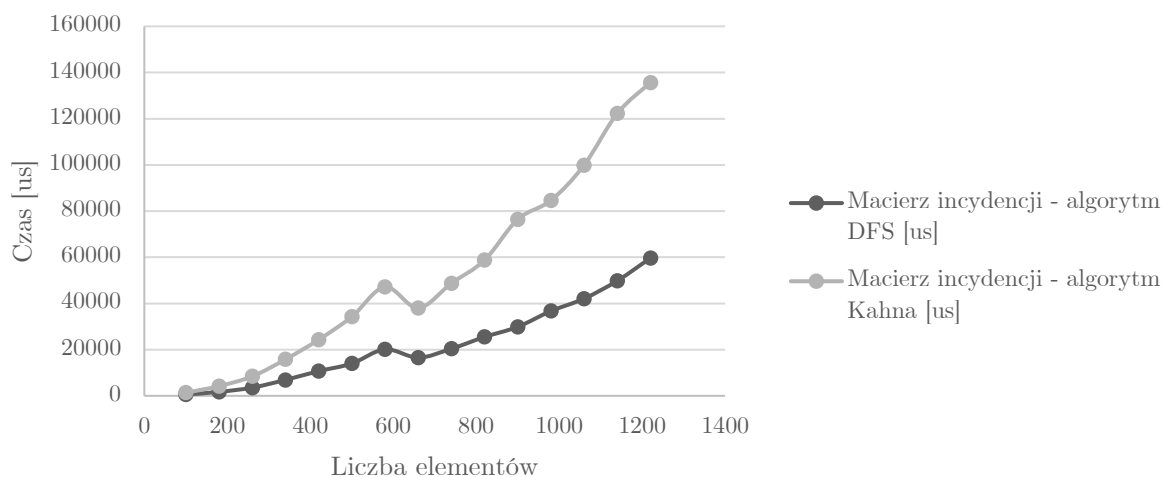
Czas został zmierzony dla piętnastu różnych grafów, gdzie liczba wierzchołków to następująco: 100, 180, 260, 340, 420, 500, 580, 660, 740, 820, 900, 980, 1060, 1140, 1220

Pomiary zostały wykonane z dokładnością do mikrosekund. Pominęto czas generowania danych wejściowych.

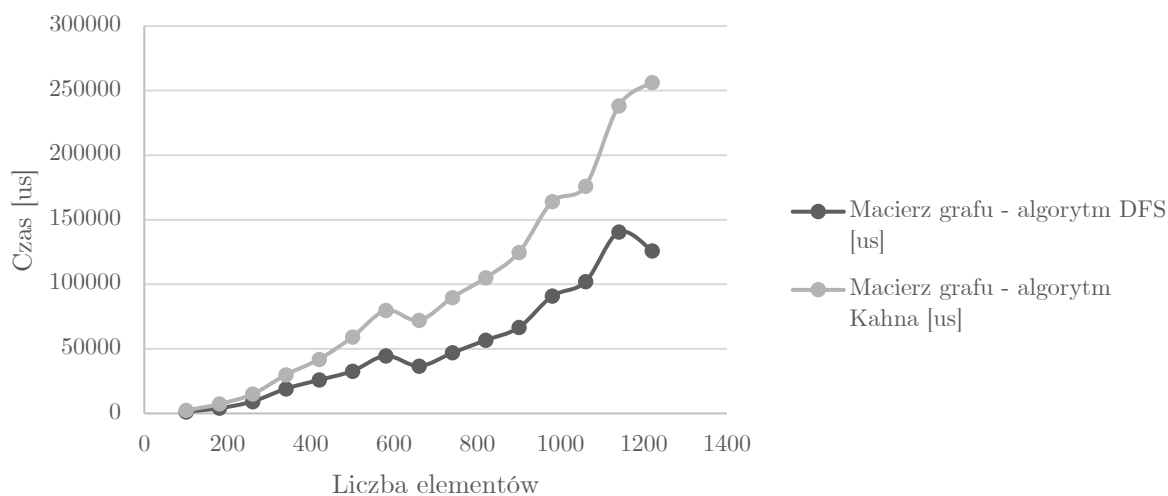
## 2 WYDAJNOŚĆ SORTOWANIA TOPOLOGICZNEGO Z WYKORZYSTANIEM DFS ORAZ ALGORYTMU KAHNA

---

Sortowanie topologiczne macierzy incydencji



Sortowanie topologiczne macierzy grafu

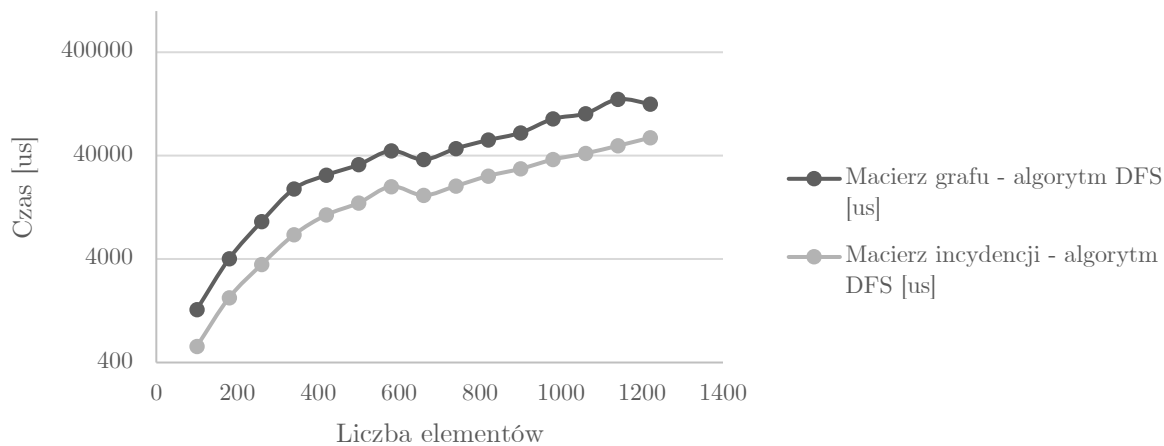


Złożoność czasowa sortowania topologicznego to  $O(V + E)$ , gdzie  $V$  to liczba wierzchołków, natomiast  $E$  to liczba krawędzi. Wydajność teoretyczna pokrywa się z testami, ponieważ liczba krawędzi w grafie z nasyceniem 50% rośnie kwadratowo względem liczby wierzchołków. Drobne rozbieżności wynikają prawdopodobnie z nierównego poziomu obciążenia systemu w trakcie wykonywania testów. Możemy zauważyć, że algorytm DFS sprawuje się lepiej od algorytmu Kahna przy tej konkretnej implementacji oraz strukturze grafu acyklicznego.

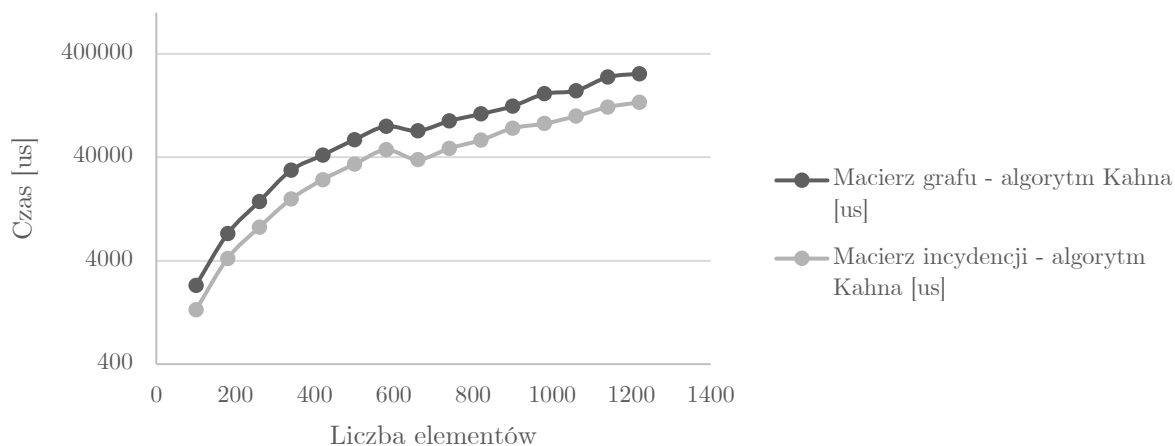
### 3 WYDAJNOŚĆ MACIERZY SĄSIEDZTWA I MACIERZY GRAFU

---

Sortowanie topologiczne algorytmem DFS



Sortowanie topologiczne algorytmem Kahna



Możemy zauważyć, że macierz grafu sprawuje się w każdym przypadku gorzej od macierzy incydencji, zajmując około dwukrotnie więcej czasu na posortowanie topologiczne danego grafu. Algorytm sortowania topologicznego jest identyczny dla obu reprezentacji grafu – różni się wyłącznie warunkiem wykrywania krawędzi na podstawie wartości w odpowiedniej kolumnie i rzędzie.

## 4 WNIOSKI

---

Sortowanie topologiczne możemy zrealizować za pomocą wielu algorytmów, w tym za pomocą algorytmu Kahna oraz wykorzystującego DFS. Oba algorytmy wykazały się dobrą wydajnością – liniową w zależności od sumy wierzchołków i krawędzi. Testy zostały przeprowadzone na dwóch reprezentacjach grafu – macierzy incydencji oraz macierzy grafu.

Należy podkreślić, że macierz grafu wykazała się gorszą wydajnością od macierzy incydencji w przetestowanych algorytmach. Ponadto jest bardziej złożona pamięciowo w porównaniu macierzy incydencji, którą można zakodować binarnie – reprezentacja poszczególnej komórki wymaga  $\log_2(3n)$  bitów pamięci, gdzie  $N$  wynosi liczbę wierzchołków. Dodatkowo tworzenie macierzy grafu jest niewydajne pod kątem czasu wykonywania oraz zużytej pamięci ze względu na konieczność utworzenia trzech oddzielnych list. Warto również wspomnieć, że macierz grafu nie jest podatna na zmiany – dodanie nowego elementu lub nowej krawędzi wymaga rekalkulacji macierzy. Tym samym, w kontekście zastosowania sortowania topologicznego macierz grafu jest znacznie gorszym wyborem od macierzy incydencji ze względu na dłuższy czas wykonywania, zwiększone zużycie pamięci oraz skomplikowany sposób implementacji.

## 5 SPIS TREŚCI

---

1	Metodologia pomiaru wydajności algorytmów.....	1
2	Wydajność sortowania topologicznego z wykorzystaniem DFS oraz algorytmu Kahna.....	2
3	Wydajność macierzy sąsiedztwa i macierzy grafu.....	3
4	Wnioski .....	4