

# Sprawozdanie

## Algorytmy grafowe

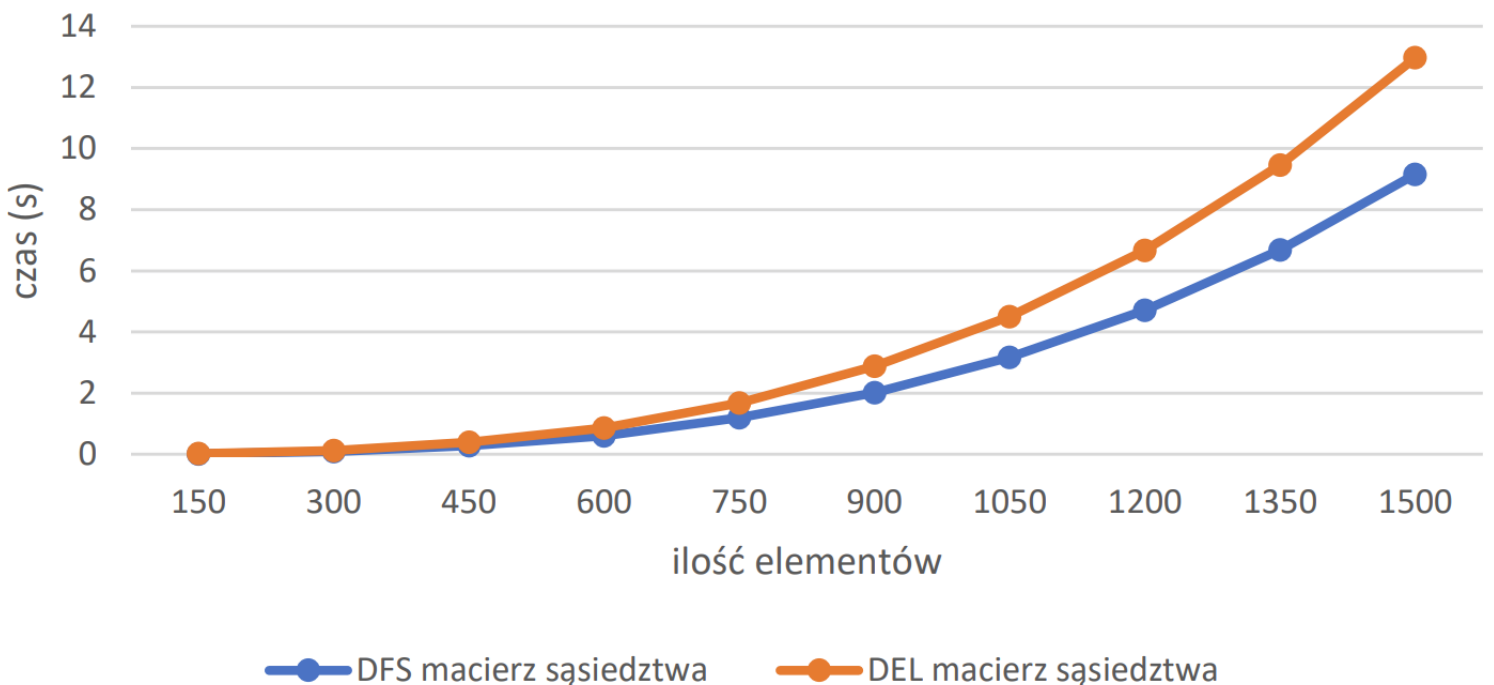
Mikołaj Pluta

1) Celem ćwiczenia jest porównanie dwóch rodzajów reprezentacji maszynowych grafów, tj. macierzy sąsiedztwa oraz listy następników oraz dwóch metod sortowania topologicznego wierzchołków – algorytm oparty na metodzie DFS oraz algorytm Kahna. Cały program zaimplementowany został w języku Python.

2) Porównanie czasów wykonywania różnych algorytmów sortowania dla danej reprezentacji maszynowej grafu.

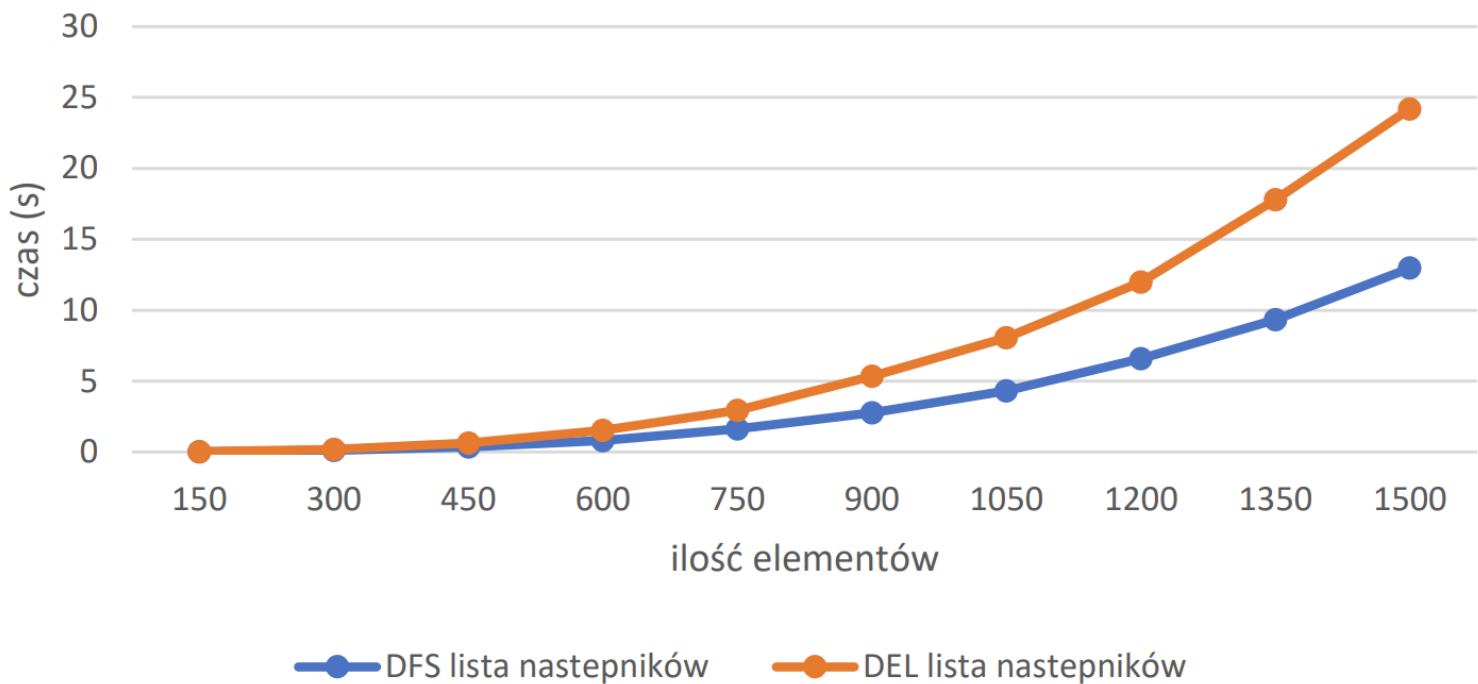
2.1) Macierz sąsiedztwa.

Wykres 1



## 2.2) Lista następników.

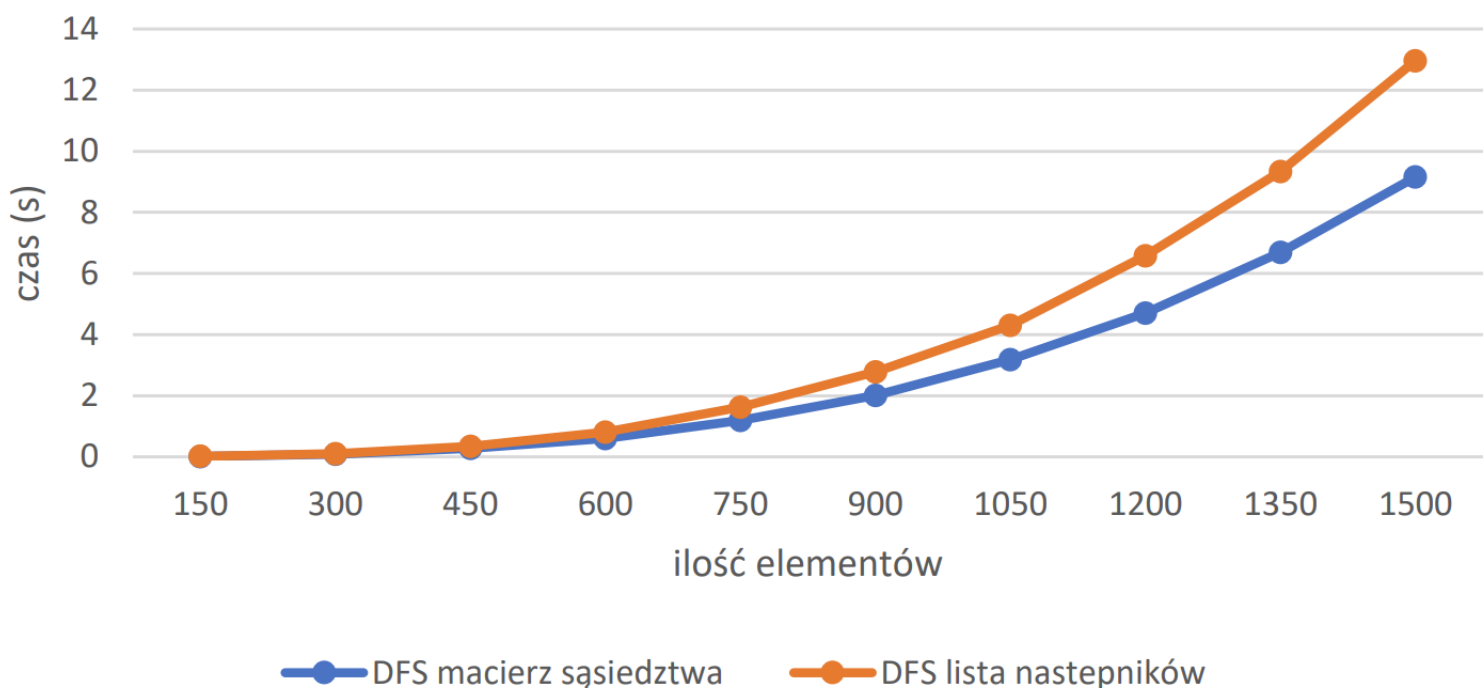
Wykres 2



## 3) Porównanie czasów wykonywania tych samych algorytmów sortowania dla różnych reprezentacji maszynowych grafu.

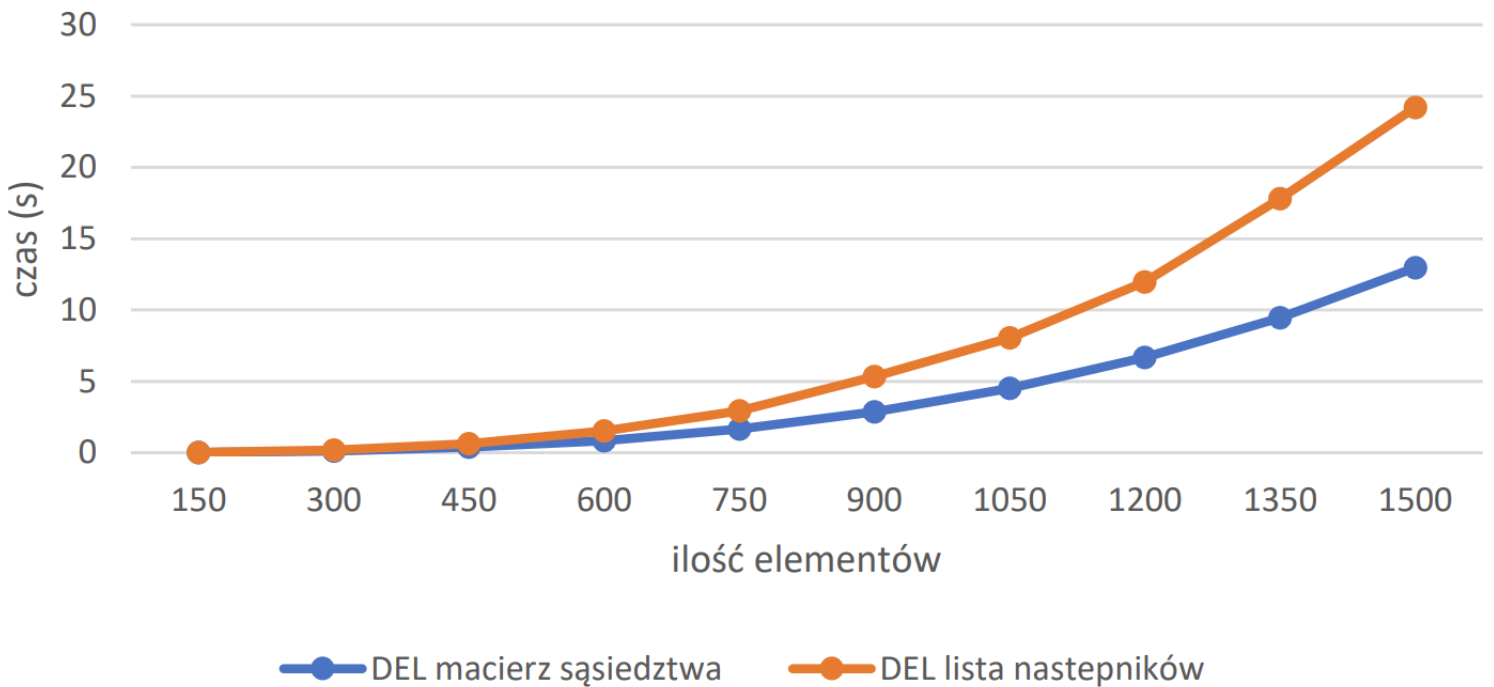
### 3.1) Sortowanie na podstawie metody DFS.

Wykres 3



### 3.2) Sortowanie algorytmem Kahna.

Wykres 4



#### 4) Podsumowanie.

4.1) Dla każdej reprezentacji maszynowej grafu zauważyć można przewagę algorytmu DFS nad algorytmem Kahna.

4.2) Pomimo liniowej złożoności czasowej przechodzenia przez wszystkie krawędzie w grafie, lista następników okazuje się nieoptymalna dla algorytmów sortowania topologicznego grafów. Wynika to między innymi z faktu, że znalezienie wierzchołka, który ma zerowy stopień wejściowy (co wymagane jest czasem także w trakcie sortowania, nie tylko na początku) wymaga w najgorszym wypadku przejścia całej listy dla każdego wierzchołka z osobna co daje złożoność kwadratową.

4.3) Wydajność powyższych algorytmów zależy również od ułożenia danych wejściowych. W grafach, na których testowane były algorytmy zauważyć można charakterystyczną strukturę, która powoduje, że algorytm wielokrotnie musi szukać nowego wierzchołka co jeszcze bardziej pogarsza wydajność algorytmów które oparte są na liście następników.

Ad 4.3) Przykłady losowych grafów tworzonych przez generator dla  $n = 50$  i nasycenia 50%.

