

15.05.2019

SPRAWOZDANIE  
PAMSI  
PROJEKT 2  
ALGORYTMY GRAFOWE

**Prowadzący:**

Dr Łukasz Jeleń

**Dane studenta:**

Kacper Starościak 241581

**Termin zajęć:**

Środa 11:15

## 1. Zadanie

Celem zadania było zaimplementowanie w języku C++ grafu w dwóch postaciach: macierzy sąsiedztwa oraz listy sąsiedztwa, oraz zbadania wydajności wybranego algorytmu grafowego.

Wybrany został algorytm Dijkstry. Wykonano pomiary czasu dla różnych reprezentacji (lista, macierz), różnych gęstości grafu (25%, 50%, 75%, 100%), oraz różnych ilości wierzchołków (10,50,100,500,1000)

## 2. Opis algorytmu

Algorytm Dijkstry jest jednym z podstawowych algorytmów grafowych. Znajduje on najkrótszą ścieżkę między dowolnymi dwoma wierzchołkami w grafie spójnym ważonym oraz oblicza jej koszt. Teoretycznie, przy implementacji kolejki jako kopiec złożoność czasowa algorytmu powinna wynosić  $O(E \cdot \log V)$  gdzie  $E$  to liczba krawędzi, a  $V$  liczba wierzchołków. Złożoność pamięciowa wynosi  $O(n^2)$  dla reprezentacji macierzowej i  $O(E+V)$  dla reprezentacji listowej.

## 3. Przebieg eksperymentu

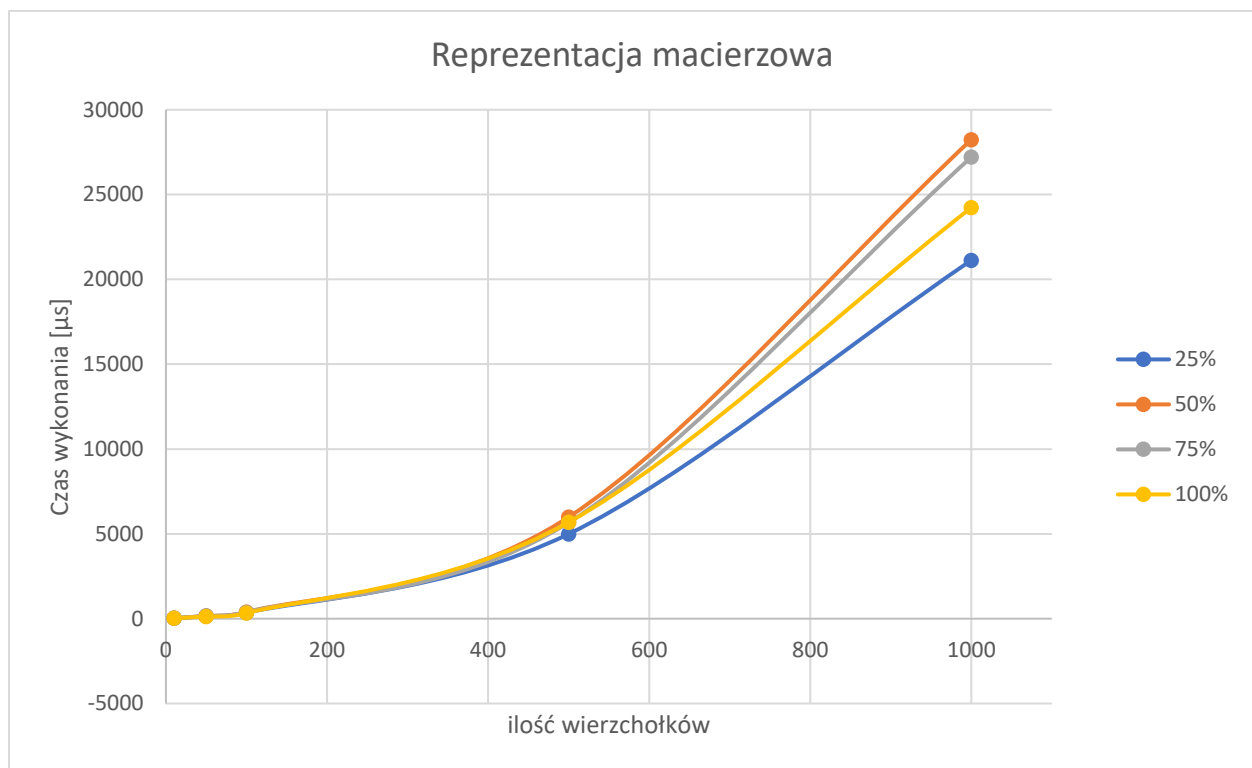
Do implementacji algorytmów w języku C++ wykorzystano program Microsoft Visual Studio 2017. Poniżej przedstawiono tabelę ze średnimi czasami działania algorytmu w różnych warunkach (dla różnych parametrów badanych grafów) oraz wykresy typu pierwszego i drugiego, zgodnie z instrukcją do zadania.

Tabela 1. Wyniki pomiaru czasu działania algorytmu przy reprezentacji macierzowej

n\rodzaj tabeli	10	50	100	500	1000
25%	25.4	153.84	357.27	4992.6	21126.68
50%	24.42	168.03	389.37	5988.61	28239.8
75%	28.53	155.11	379.82	5669.14	27208.4
75%	27.84	134.49	331.7	5682.79	24231.07

Tabela 2. Wyniki pomiaru czasu działania algorytmu przy reprezentacji listowej

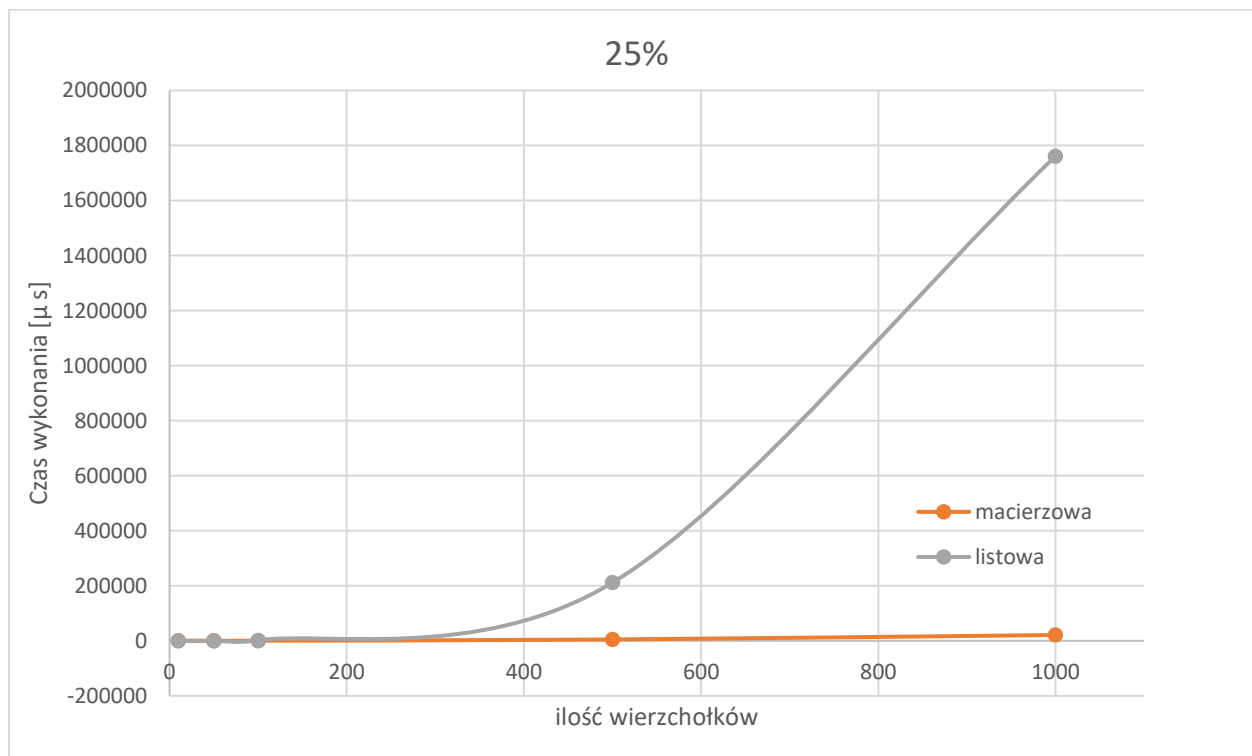
n\rodzaj tabeli	10	50	100	500	1000
25%	36.61	407.2	1716.22	211554.7	1761117
50%	39.3	555.56	3112.11	469313.2	4107232
75%	44.06	726.18	5171.91	715614.9	6292449
75%	46.06	993.95	6540.09	1031292	8668847



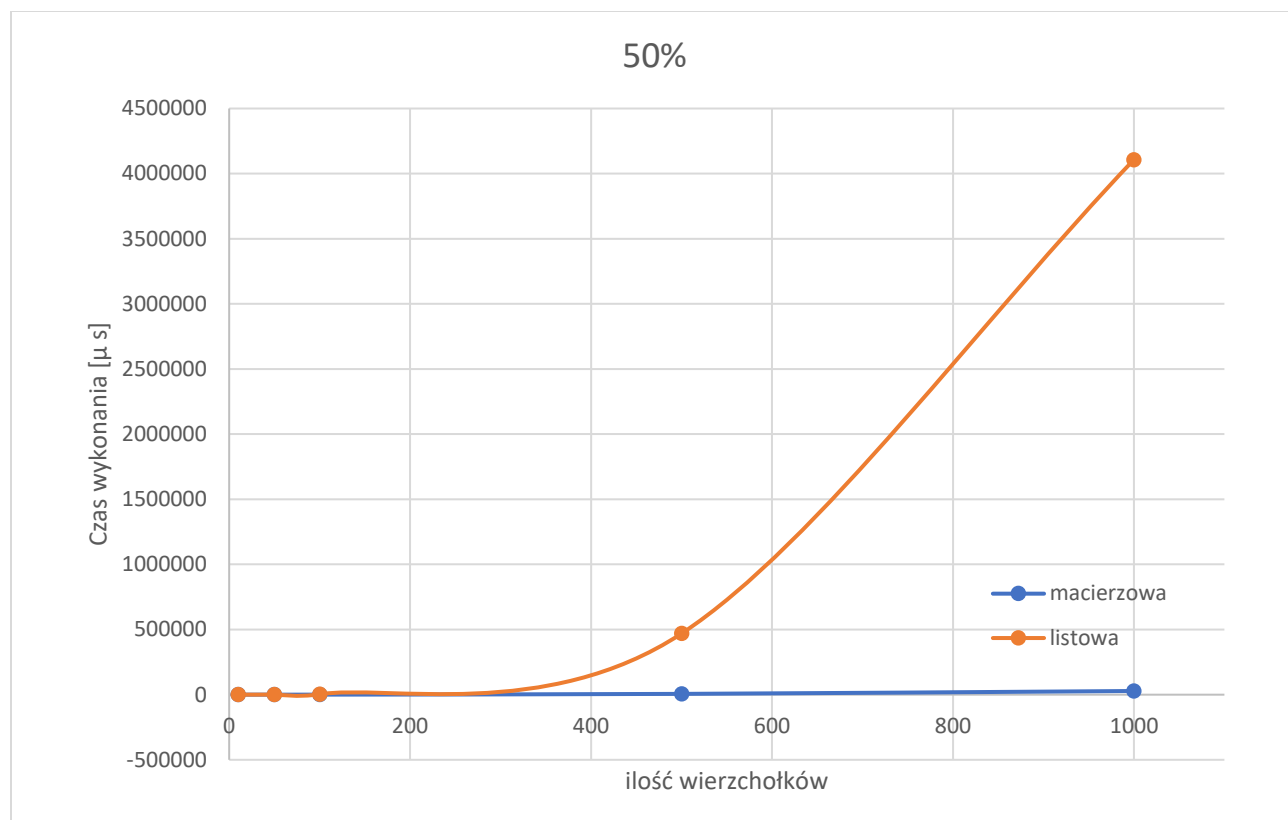
Wykres 1. Wyniki pomiarów czasu dla reprezentacji macierzowej



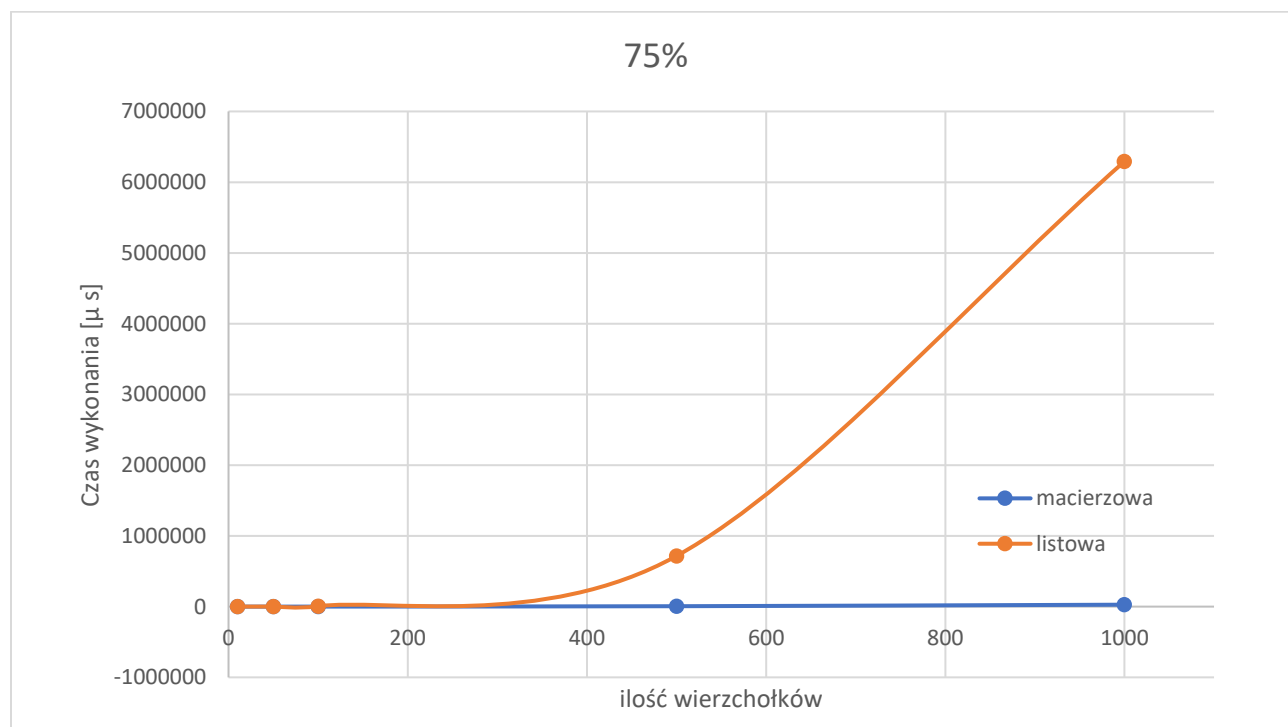
Wykres 2. Wyniki pomiarów czasu dla reprezentacji listowej



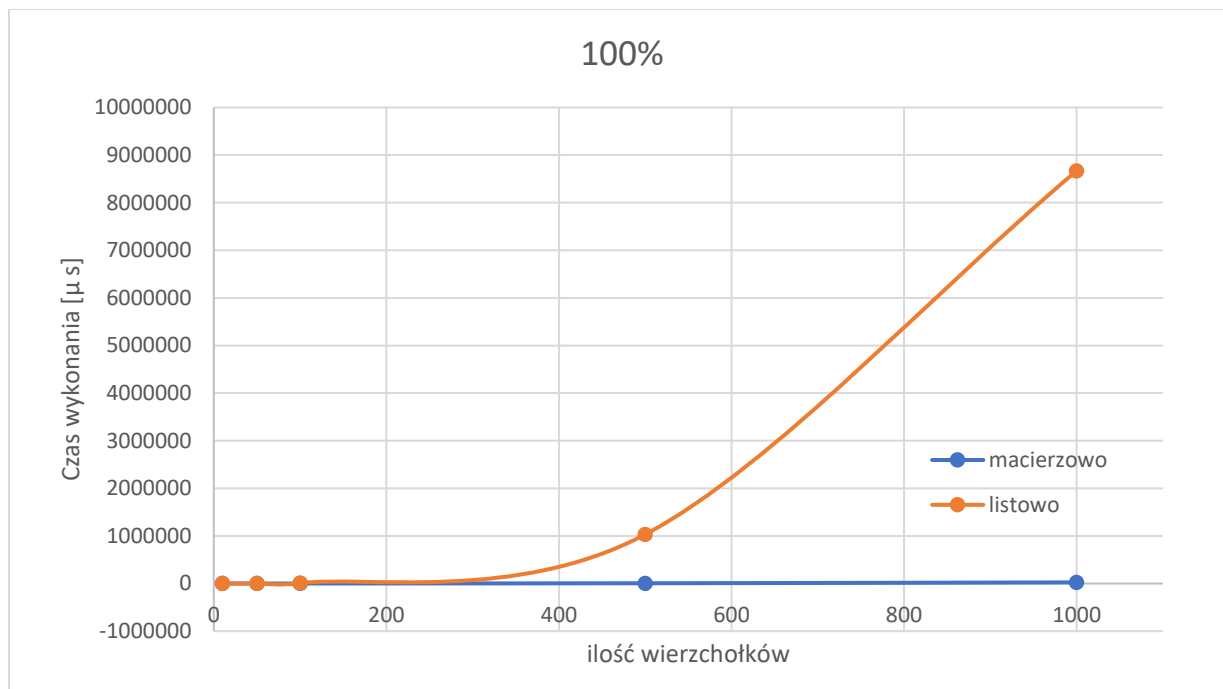
Wykres 3. Wyniki pomiarów czasu dla gęstości 25%



Wykres 4. Wyniki pomiarów czasu dla gęstości 50%



Wykres 5. Wyniki pomiarów czasu dla gęstości 75%



Wykres 6. Wyniki pomiarów czasu dla gęstości 100%

#### 4. Wnioski

Po raz kolejny wyniki są zadziwiające. Można by się spodziewać, że reprezentacja listowa grafu będzie wydajniejszym rozwiązaniem, niż jej macierzowy odpowiednik, jednakże wyniki na to nie wskazują. Prawdopodobnie wynika to z niedociągnięć w kodzie i niepotrzebnie wykonywanych operacji.

Wykresy typu drugiego z tego powodu są nieczytelne i trudne jest porównanie wartości czasu wykonywania algorytmu dla obydwu reprezentacji.

Kształty wykresów typu pierwszego przypominają w przybliżeniu krzywe teoretyczne wynikające ze złożoności obliczeniowej algorytmu. Widać na nich zależność czasu od zwiększającej się gęstości.

- Źródła
- [https://pl.wikipedia.org/wiki/Algorytm\\_Dijkstry](https://pl.wikipedia.org/wiki/Algorytm_Dijkstry)
- [https://eduinf.waw.pl/inf/alg/001\\_search/0138.php](https://eduinf.waw.pl/inf/alg/001_search/0138.php)