

Data oddania projektu:
08.05.2019

Prowadzący:
mgr inż. Marcin Ochman
Wykonawca:
Julia Krzeszowska 241615

SPRAWOZDANIE

PROJEKT nr 3 - Grafy

1. Cel projektu

Zapoznanie się z algorytmem Dijkstrty oraz algorytmem Bellmana-Forda służącymi do rozwiązywania problemu najkrótszej drogi (ścieżki) w grafie.

2. Przebieg projektu

Został napisany program, w którym zostały zaimplementowane wyżej wymienione algorytmy dla dwóch sposobów przechowywania informacji o wierzchołkach w grafie.

W projekcie znajdują się następujące klasy:

- klasa *NeighbourList*, która implementuje listę sąsiadów
- klasa *NeighbourMatrix*, która implementuje macierz sąsiedztwa
- klasa *Pair*, która przechowuje parę dowolnych obiektów
- klasa *Timer* służąca do pomiaru czasu
- klasa *Tests* służąca do pojedynczych testów obu algorytmów
- klasa *Algorithms*, w której zostały zaimplementowane algorytmy wyszukiujące

Dodatkowo w projekcie znajduje się plik *main.cpp*, w którym wykonano zadane testy dla wybranego algorytmu (algorytm Bellmana-Forda).

Dla pięciu różnych liczb wierzchołków $V = \{50, 100, 150, 200, 250\}$ oraz gęstości grafu $E = \{25\%, 50\%, 75\%, 100\%\}$ została zbadana efektywność algorytmu Bellmana-Forda. Wygenerowano po 100 losowych instancji, a wyniki uśredniono. Poniższa tabela zawiera uśrednione wyniki czasu działania algorytmu dla różnej liczby wierzchołków oraz gęstości grafu. W tabeli *Tab.1* zamieszczono wyniki dla listy sąsiadów, a w tabeli *Tab.2* dla macierzy sąsiedztwa. Czasy podane są w sekundach.

	50	100	150	200	250
25%	0,006533	0,087111	0,0425149	1,410529	3,536144
50%	0,02125	0,320066	1,67996	5,775814	14,449422
75%	0,043874	0,699003	3,903159	13,310085	58,815426
100%	0,074871	1,258657	7,132448	23,888158	66,647557

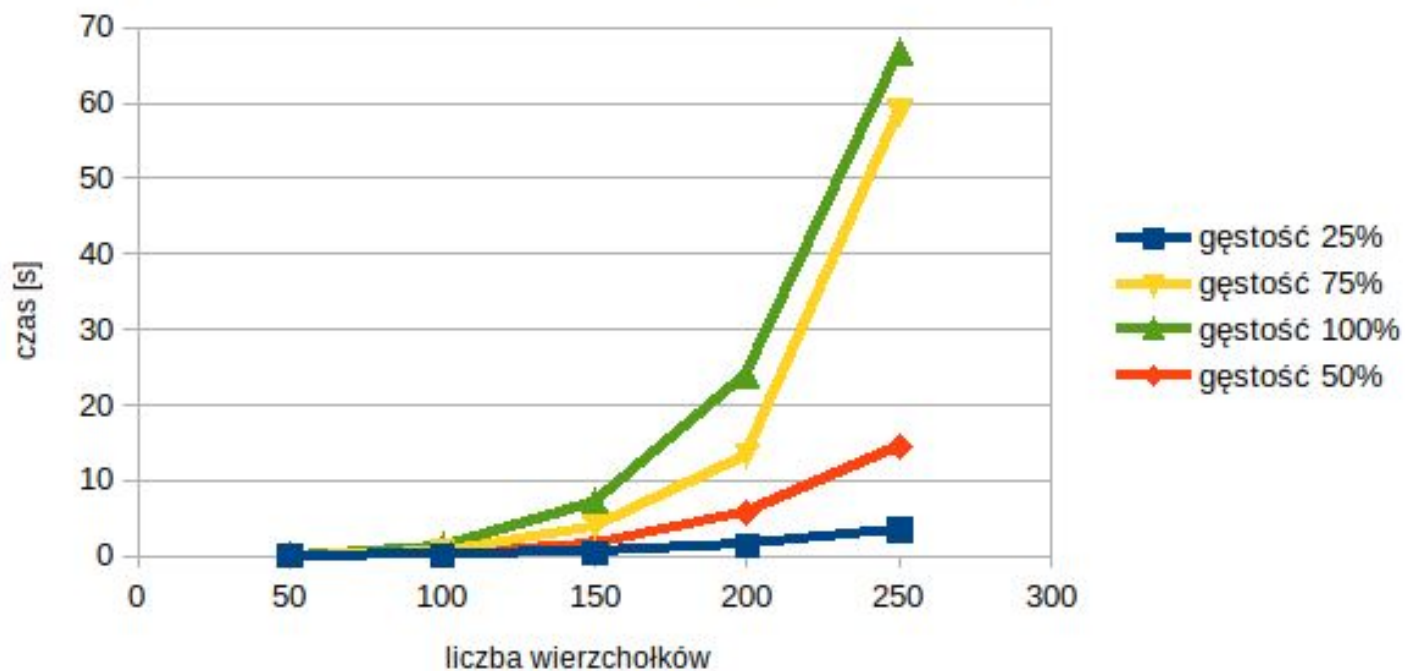
Tab.1

	50	100	150	200	250
25%	0,000748	0,005861	0,019639	0,047069	0,09322
50%	0,00148	0,011868	0,039259	0,093598	0,183848
75%	0,002193	0,017398	0,060168	0,139411	0,274036
100%	0,002904	0,023254	0,077999	0,185345	0,362434

Tab.2

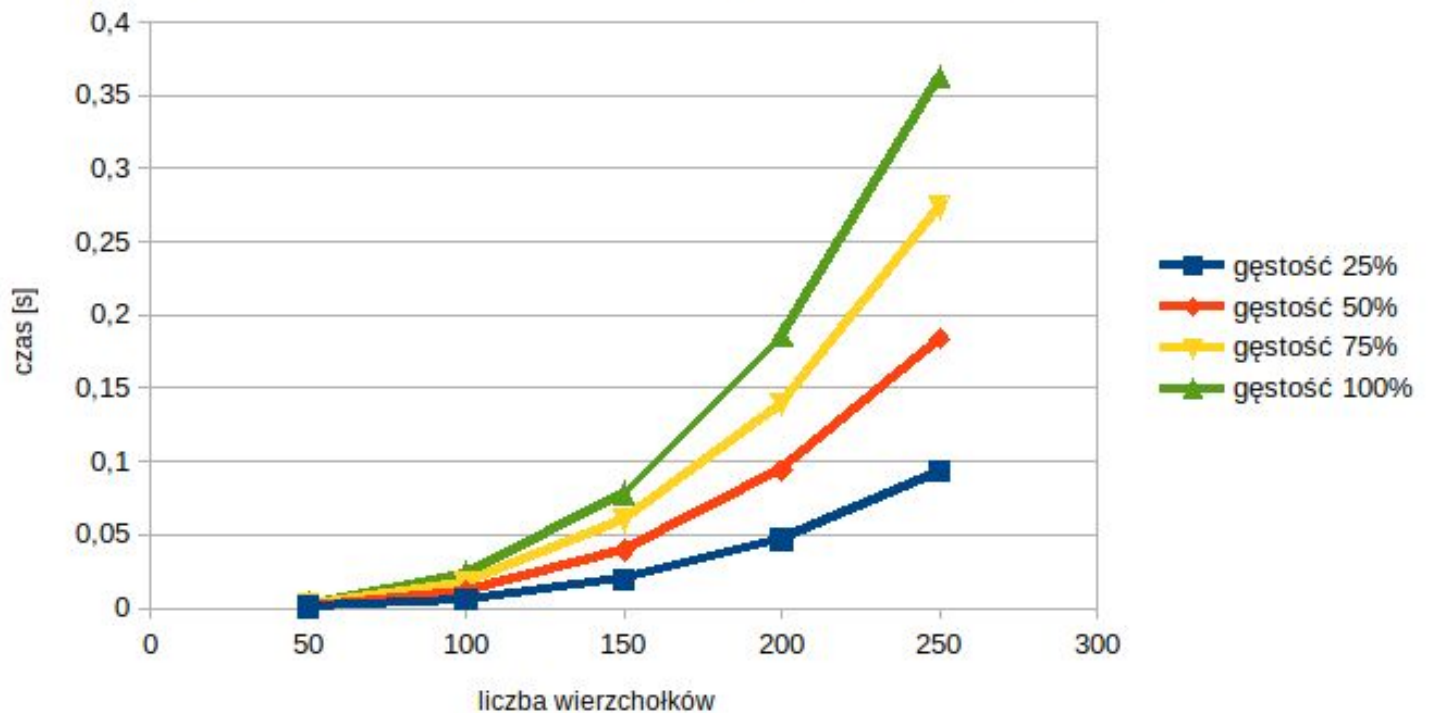
Na podstawie Tab.1 i Tab.2 wygenerowano wykresy Wyk.1 i Wyk.2.

Wykres zależności czasu trwania algorytmu od liczby wierzchołków



Wyk.1 - wykres dla listy sąsiadów

Wykres zależności czasu trwania algorytmu od liczby wierzchołków



Wyk.2 - wykres dla macierzy sąsiedztwa

3. Wnioski

Zaobserwowano, że algorytm Bellmana-Forda działa dużo szybciej dla macierzy sąsiedztwa niż dla listy sąsiadów. Z wykresów można odczytać, że złożoność obliczeniowa algorytmu zgadza się ze złożonością podaną w literaturze, tj. $O(VE)$.