Sprawozdanie z zajęć laboratoryjnych PAMSI

Grafy

1. Cel zadania

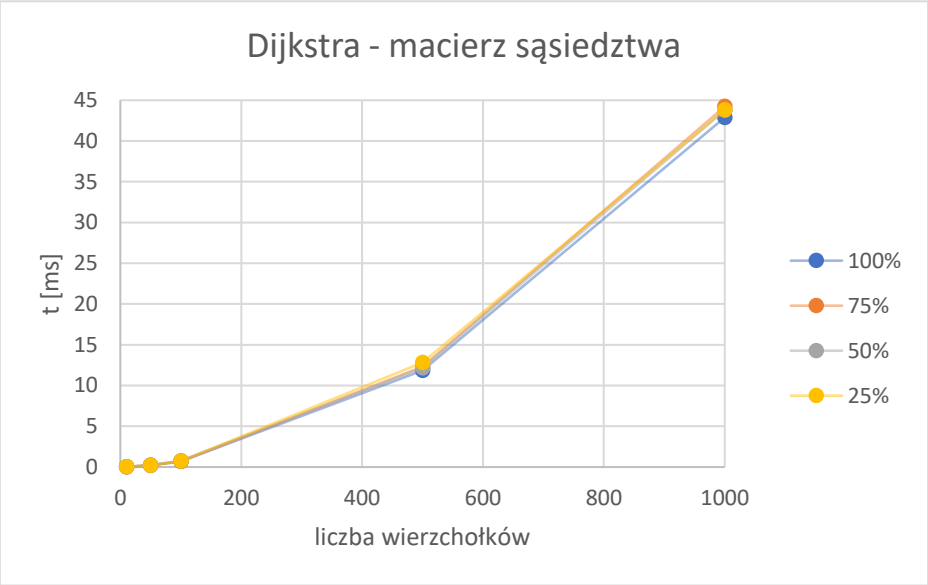
Zapoznanie się z reprezentacją grafów oraz dwoma algorytmami rozwiązującymi problem znalezienia najkrótszej ścieżki w grafie tj. algorytm Bellmana-Forda oraz algorytm Dijkstry.

2. Badania zależności czasu działania algorytmów

Wyniki ponad 50 pomiarów czasu działania wyżej wymienionych algorytmów dla dwóch reprezentacji grafu (macierzy oraz listy sąsiedztwa) znajdują się w repozytorium w pliku „pomiary.xlsx”

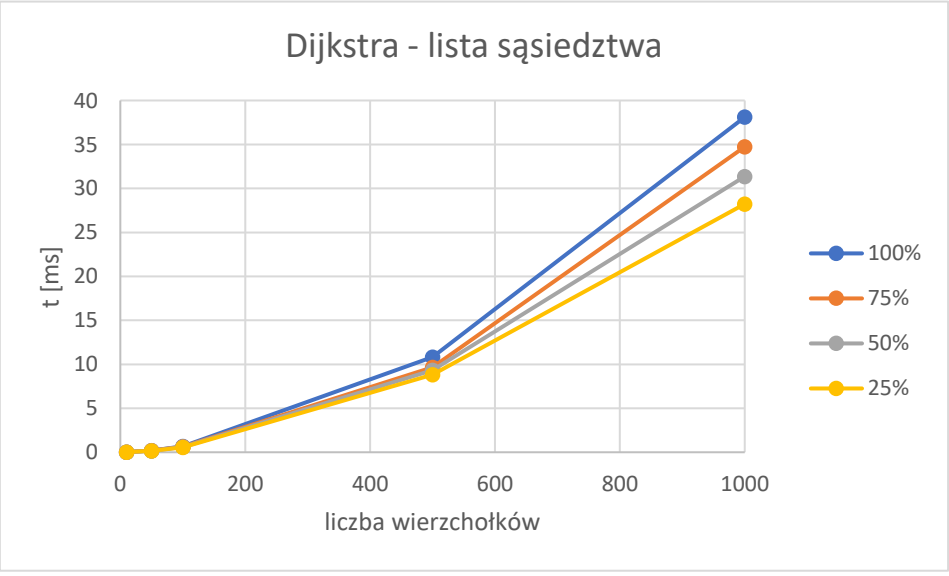
Uśrednione wyniki pomiarów przedstawione zostały za pomocą wykresów dla czterech różnych przypadków:

A) Algorytm Dijkstry – Macierz sąsiedztwa



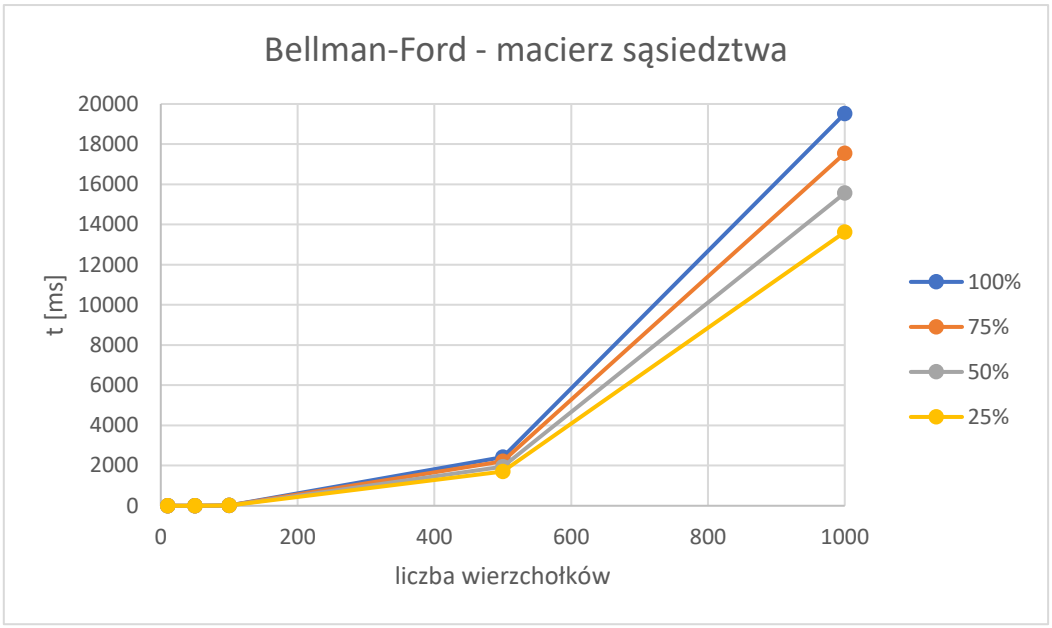
Rysunek 1 – Dijkstra - macierz

B) Algorytm Dijkstry – lista sąsiedztwa



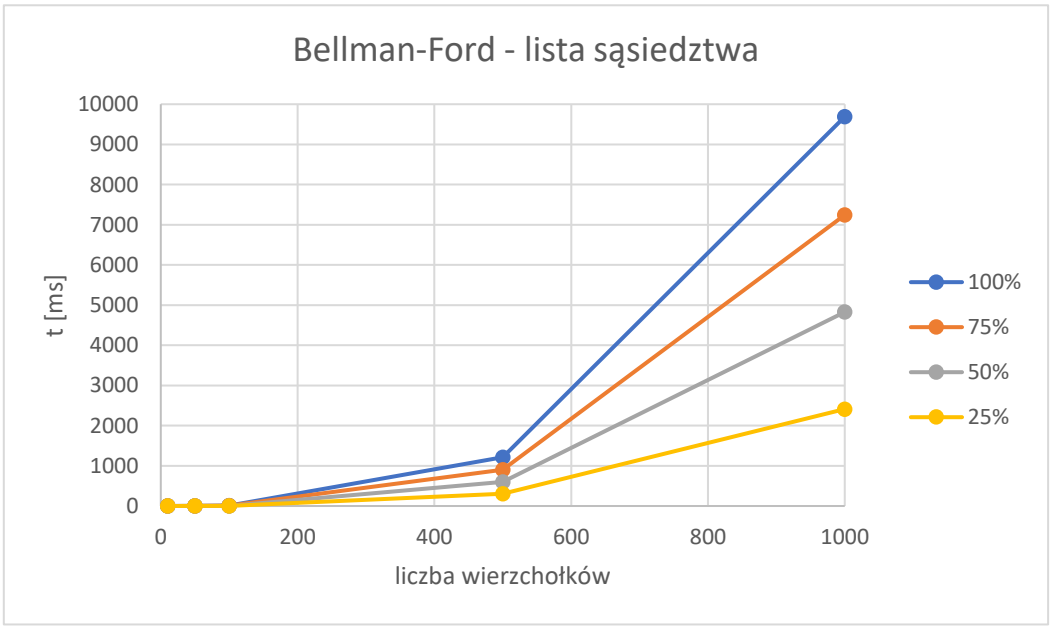
Rysunek 2 – Dijkstra - lista

C) Algorytm Bellmana-Forda – macierz sąsiedztwa



Rysunek 3 – Bellman-Ford - macierz

D) Algorytm Bellmana-Forda – lista sąsiedztwa



Rysunek 4 – Bellman-Ford - lista

3. Wnioski

Porównanie badanych algorytmów przedstawia poniższa tabela:

	Djikstra	**Bellman – Ford**
Dodatnie koszty	Działa poprawnie	Działa poprawnie
Dodatnie i ujemne koszty	Nie działa	Działa poprawnie dla grafów skierowanych
Ujemne cykle	Nie wykrywa	Wykrywa w grafach skierowanych
Złożoność algorytmu	O(E*log(V))	O(V*E)

Tabela 1 – Dijkstra - Bellman-Ford - porównanie

Z tabelki oraz wykresów wywnioskować można, że algorytm Dijkstry jest znacznie szybszy od algorytmu Bellmana Forda. Dla obydwu badanych algorytmów reprezentacja grafu jako listy sąsiedztwa przyniosła korzystniejsze czasy działania niż dla macierzy sąsiedztwa. Co ciekawe, gęstość grafu ma znikomy wpływ na czas działania algorytmu Dijkstry z macierzą sąsiedztwa. Jeśli nie mamy pewności wystąpienia wyłącznie dodatnich kosztów przejścia, powinniśmy zdecydować się na użycie algorytmu Bellmana-Forda, który obsługuje grafy skierowane z ujemnymi kosztami oraz wykrywa ujemne cykle. W przeprowadzonych przeze mnie badaniach uzyskał on jednak znacznie większe czasy działania, co może być spowodowane dodatkowym czasem wykorzystywanym na sprawdzenie ujemnych cykli. Podsumowując, z badanych konfiguracji najkorzystniej wypada algorytm Dijkstry z listą sąsiedztwa, jednak wybór tego algorytmu musi wiązać się z wiedzą na temat kosztów przejścia w danym grafie.