

Data: 30.04.2020

Imię i nazwisko: Karolina Głuszek

Nr albumu: 249034

Nazwa kursu: Projektowanie algorytmów i metody sztucznej inteligencji

Dane prowadzącego: mgr Marta Emirsajłow

Termin zajęć: piątek, 9.15

1. Wprowadzenie

Analizowany oraz zaimplementowany przeze mnie algorytm znajdowania najkrótszej ścieżki w grafie to algorytm Bellmana-Forda. W celu zbadania jego efektywności wykonane zostały testy na dwóch sposobach reprezentacji grafu: jako macierzy oraz listy sąsiedztwa. Badania efektywności zostały wykonane dla kolejno 10, 50, 100, 500, 1000 wierzchołków grafu oraz dla 25%, 50%, 75% i 100% gęstości grafu. Dla każdego zestawu parametrów wygenerowano 100 losowych instancji, których uśrednione wyniki zostały umieszczone w sprawozdaniu.

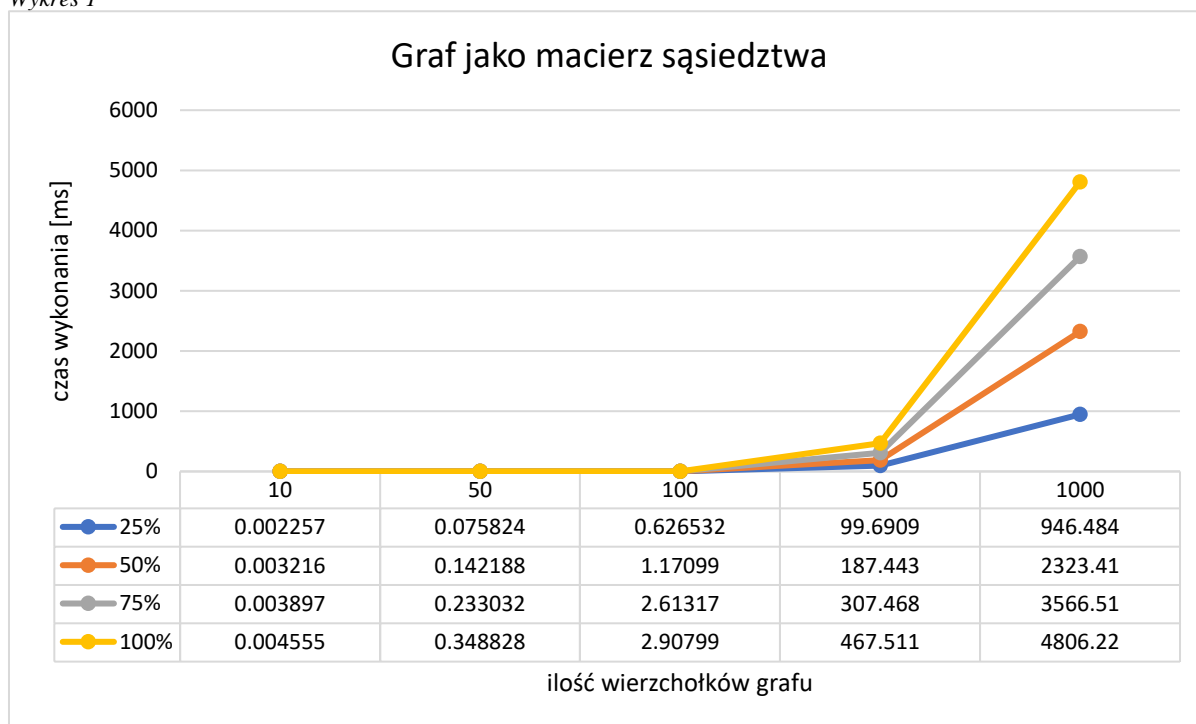
2. Opis algorytmu

Algorytm Bellmana-Forda wykorzystywany jest do znajdowania najkrótszej ścieżki od wierzchołka początkowego do wszystkich pozostałych wierzchołków w skierowanym grafie ważonym. W danym grafie mogą występować ujemne wartości wag krawędzi, ale nie są dopuszczalne ujemne cykle, które zostają wykryte przez algorytm.

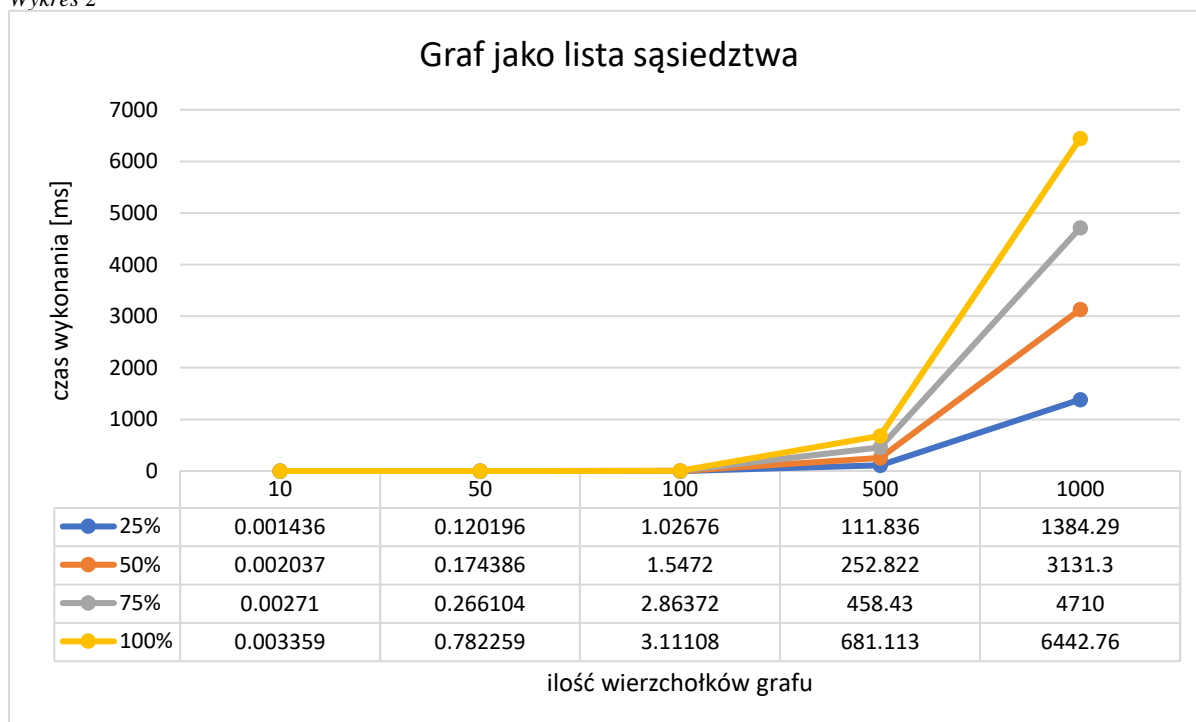
Omawiany algorytm działa w oparciu o metodę relaksacji krawędzi, która sprawdza, czy przechodząc daną krawędź do wybranego wierzchołka nie otrzymaliśmy krótszej ścieżki niż dotychczasowa. W tym celu w programie tworzona jest tablica odległości indeksowana wszystkimi wierzchołkami grafu. Na początku odległość do wierzchołka początkowego ustawiana jest na 0, a do pozostałych wierzchołków na nieskończoność. Następnie wykonywana jest relaksacja wszystkich krawędzi, których liczba wynosi $|E|$. Ponieważ najdłuższa możliwa ścieżka do danego wierzchołka ma długość $|V| - 1$, gdzie $|V|$ to liczba wierzchołków grafu, relaksacja krawędzi musi zostać wykonana $|V| - 1$ razy, aby została przeanalizowana każda możliwa ścieżka. Z tego wynika, że złożoność czasowa algorytmu Bellmana-Forda wynosi $O(|V| * |E|)$.

3. Wyniki

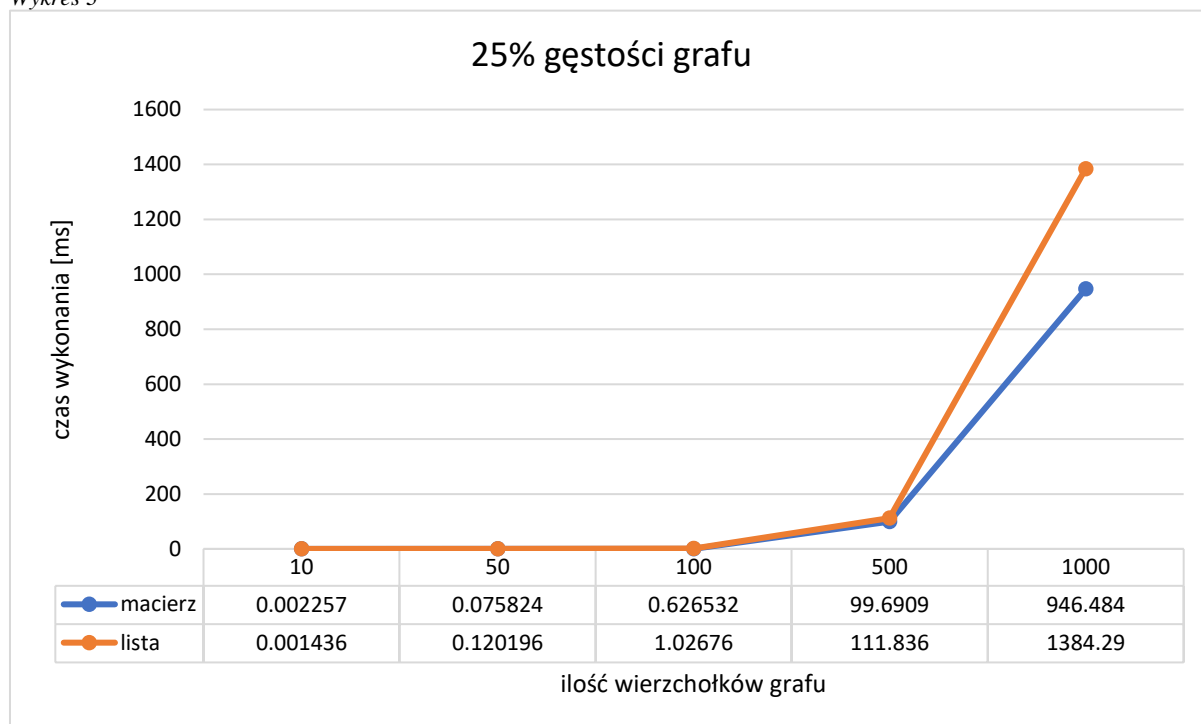
Wykres 1



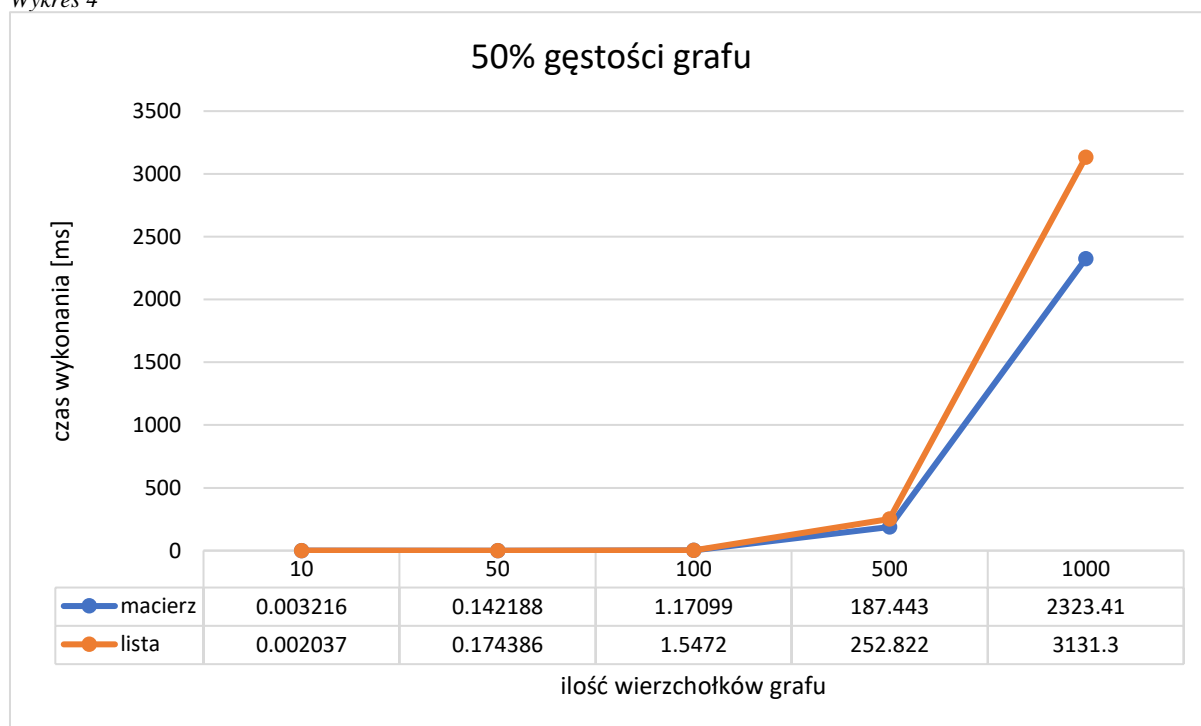
Wykres 2



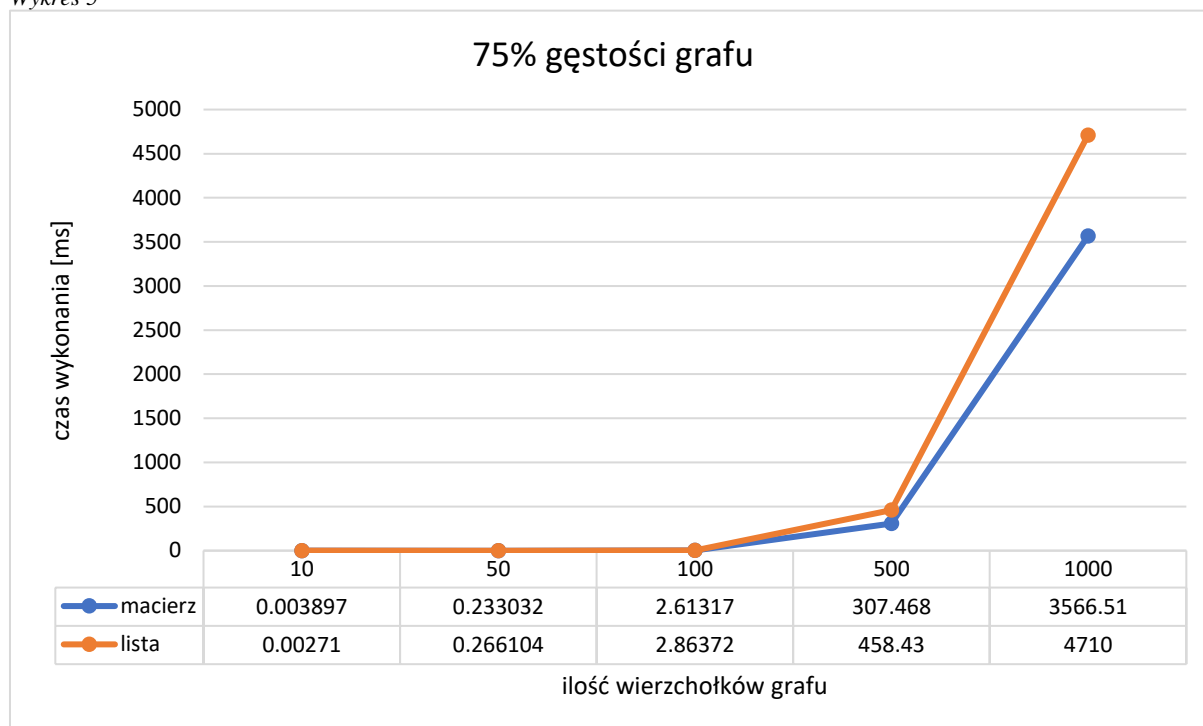
Wykres 3



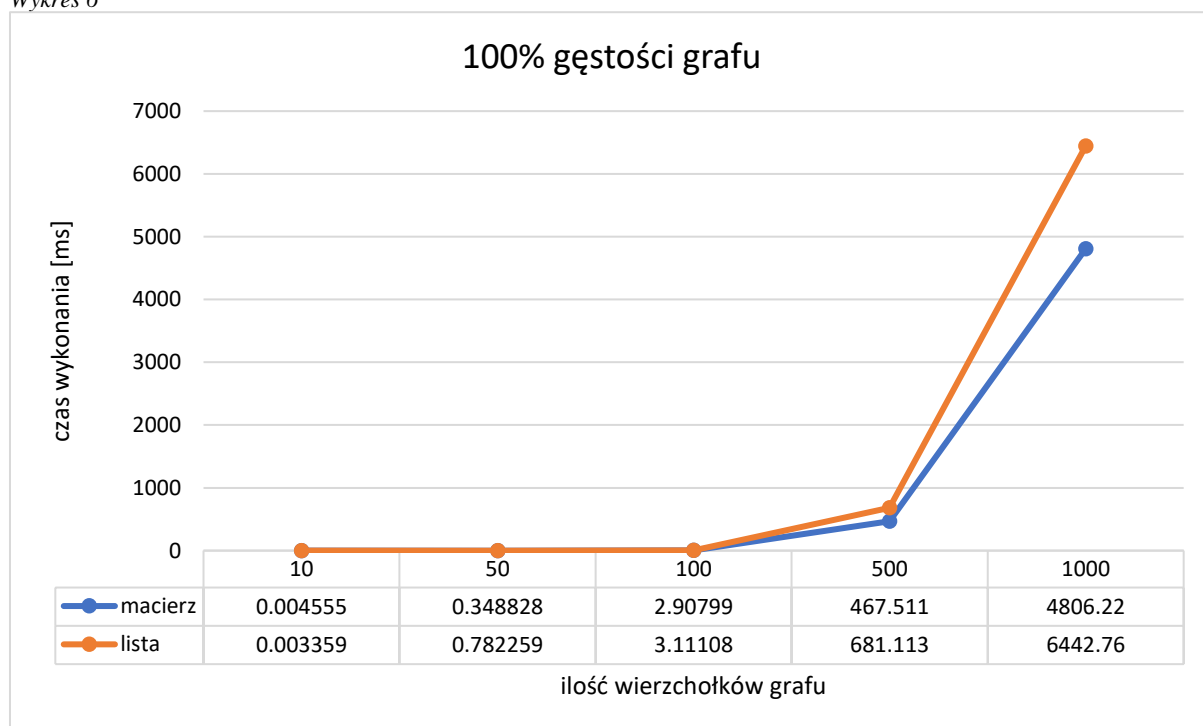
Wykres 4



Wykres 5



Wykres 6



4. Podsumowanie i wnioski

Wykresy 1 i 2 przedstawiają zależność czasu wykonania algorytmu Bellmana-Forda w funkcji ilości wierzchołków grafu, gdzie gęstość grafu jest parametrem. Osobno przedstawione są reprezentacje grafu jako macierzy oraz listy sąsiedztwa. Z wykresów tych wynika, że w obu sposobach reprezentacji, im większa była gęstość grafu i ilość wierzchołków, tym dłużej trwało wyszukanie najkrótszych ścieżek za pomocą algorytmu Bellmana-Forda.

Wykresy od 3 do 6 ilustrują zależność czasu od ilości wierzchołków dla obu reprezentacji grafu naraz, przy czym każdy wykres odpowiada jednej gęstości grafu. Na wszystkich wykresach można zauważyć, że dla najmniejszej ilości wierzchołków równej 10 algorytm wykonał się szybciej w przypadku reprezentacji grafu jako listy sąsiedztwa. Natomiast dla wszystkich pozostałych ilości wierzchołków (50, 100, 500, 1000) czas wykonania algorytmu Bellmana-Forda był mniejszy w przypadku grafu jako macierzy sąsiedztwa. Z wykresów można również odczytać, że wraz ze wzrostem ilości wierzchołków i gęstości grafu różnica w czasie między dwoma reprezentacjami rośnie na korzyść macierzy sąsiedztwa.

Mimo, że lista sąsiedztwa ma mniejszą złożoność pamięciową, a przejrzenie wszystkich krawędzi w jej grafie mniejszą złożoność czasową, macierz sąsiedztwa okazała się reprezentacją, dla której czas działania algorytmu był w prawie każdym przypadku mniejszy. Względnie duże różnice w czasie pomiędzy dwoma reprezentacjami grafu wynikają prawdopodobnie ze sposobu ich implementacji w programie. Macierz sąsiedztwa jest tworzona jako tablica dwuwymiarowa wskaźników na krawędzie grafu, podczas gdy lista sąsiedztwa jest tablicą własnych list jednokierunkowych zawierających wskaźniki na krawędzie. W związku z tym czas alokacji oraz zwalniania pamięci będzie wydłużony w przypadku listy. Dodatkowo efektywność reprezentacji macierzowej wzrasta wraz z gęstością grafu, kiedy prawie wszystkie elementy macierzy są wypełnione krawędziami.

5. Literatura

Michael T. Goodrich, "Data Structures and Algorithms in C++"

https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda

<http://www.algorytm.org/algorytmy-grafowe/algorytm-forda-bellmana.html>

http://algorytmy.ency.pl/tutorial/algorytm_bellmana_forda