

Grafy

Wykonał: Kacper Szczepanowski 249466
Projektowanie algorytmów i metody sztucznej inteligencji
wt. 11¹⁵ – 15⁰⁰

Prowadzący: Dr inż. Andrzej Rusiecki
Data: 5 maja 2020r.

1 Wstęp - Opis projektu

1.1 Cel

Celem projektu było napisanie programu z implementacją struktury grafu, a następnie implementacja algorytmu znajdującego najkrótszą ścieżkę do pozostałych wierzchołków i przeprowadzić badania tego algorytmu. Badania algorytmu Bellmana-Forda przeprowadzono dla 5 różnych liczb wierzchołków (10, 50, 100, 150, 300), dla gęstości: 25%, 50%, 75% i 100% czyli dla grafu pełnego.

1.2 Środowisko pracy

Program napisany został w środowisku Code::Blocks w języku C++. Testy przeprowadzono na systemie Linux Debian (GNOME) zainstalowanym na dysku HDD (dual-boot obok systemu Windows 11), na komputerze o parametrach: Intel Core i7-7700HQ 2,8 GHz i 16GB RAM. Czas badano przy pomocy biblioteki chrono.

2 Opis działania algorytmu

Jest to algorytm służący do wyszukiwania najkrótszych ścieżek w grafie ważonym. Metoda działania opiera się na relaksacji, czyli sprawdzeniu czy po przejściu daną krawędzią nie zmniejszy się koszt drogi. Algorytm może też służyć, do wynajdywania ujemnych cykli, bo (w odróżnieniu od alg. Dijkstry) może być wykorzystywany przy ujemnych wagach. Złożoność czasowa: $O(|V| \cdot |E|)$; gdzie V-liczba wierzchołków, E - liczba krawędzi.

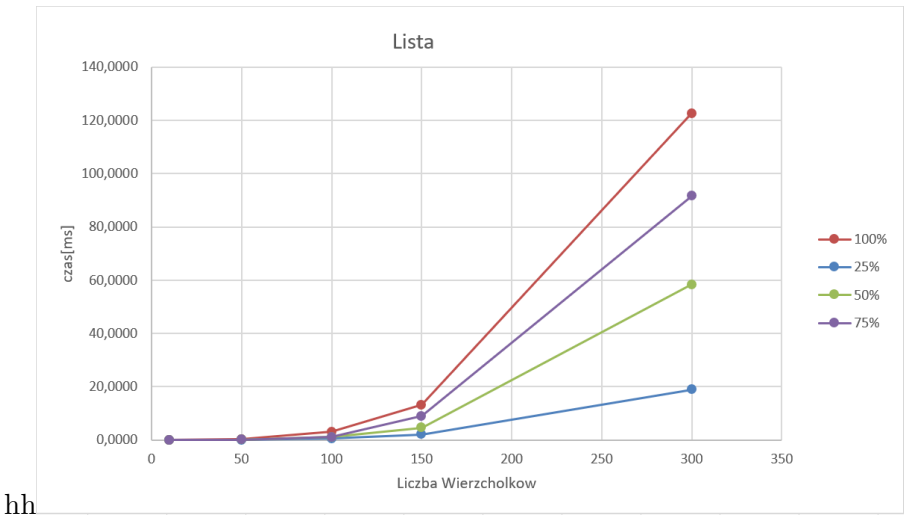
3 Badania

3.1 Tabele

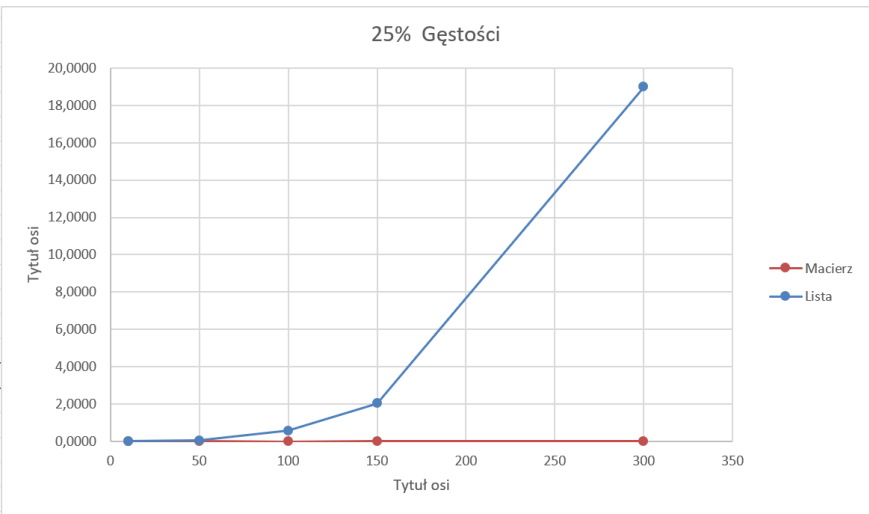
Macierz					
czas\L.wierz	10	50	100	150	300
25%	0,00026235	0,0002505	0,00024298	0,00044872	0,00086319
50%	0,0001402	0,0001621	0,00026405	0,0004428	0,00100622
75%	0,0000926	0,0001844	0,00031378	0,0004813	0,00104241
100%	0,0000751	0,0001448	0,00028895	0,00045267	0,00105103

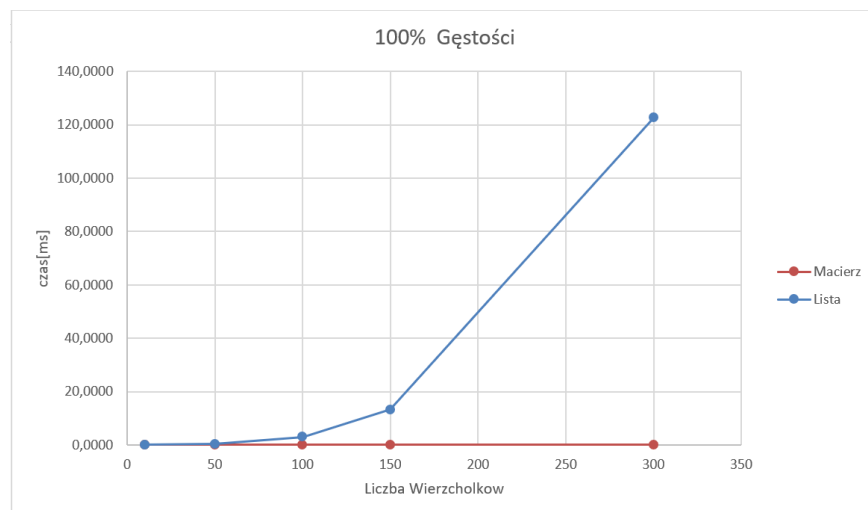
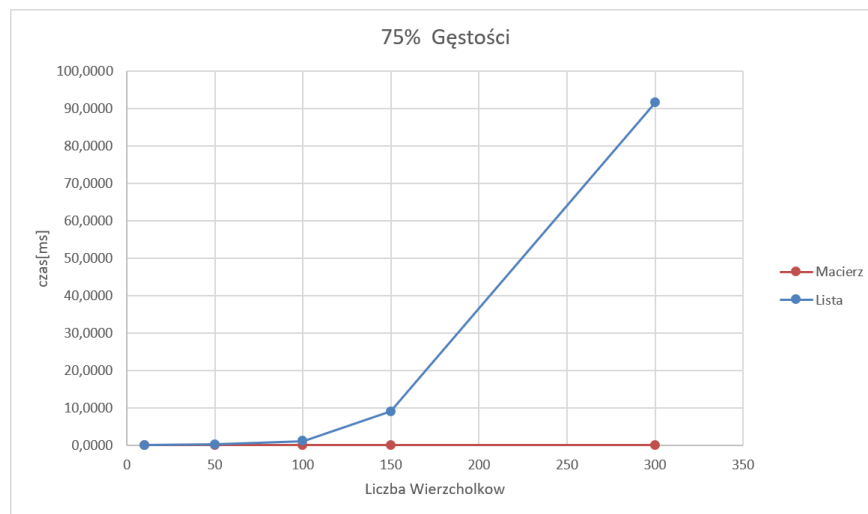
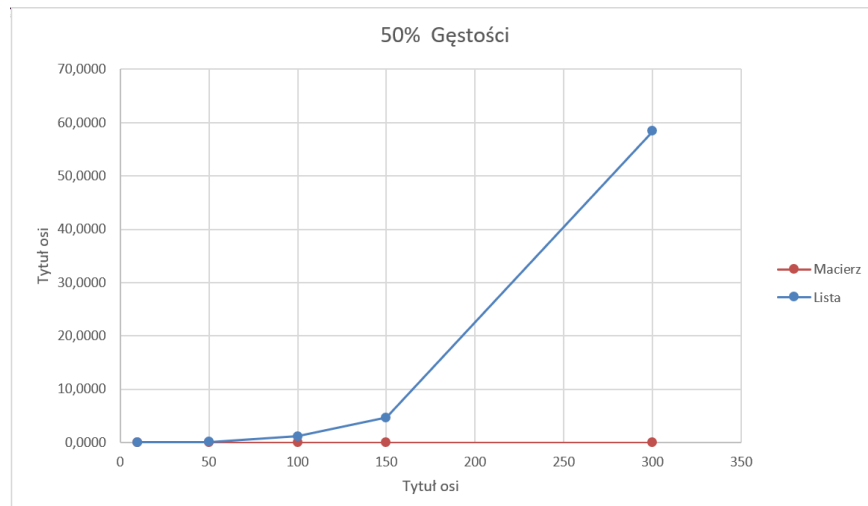
Lista					
czas\L.wierz	10	50	100	150	300
25%	0,0014508	0,053214	0,57828	2,0329	18,9846
50%	0,00250935	0,136802	1,1748	4,68246	58,4095
75%	0,00120212	0,20736	1,176903	9,0079	91,5969
100%	0,00122011	0,30223	3,03633	13,2402	122,564

3.2 Wykresy Typ 1



3.3 Wykresy typ 2





4 Wnioski

Algorytm Bellmana-Forda o wiele szybciej działa gdy graf zaimplementowany jest na macierzy sąsiedztwa. Dla każdego stopnia gęstości grafu czas wykonywania się algorytmu na liście sąsiedztwa jest o wiele większy. Dla macierzy dla gęstości grafu równej 25% czas wykonywania algorytmu BF jest mniejszy dla 100 wierzchołków niż dla 25 i 50, co może budzić pewne obawy. Potwierdziły się przypuszczenia początkowe, że graf zaimplementowany na macierzy potrzebuje

mniej czasu na wykonanie się badania ponieważ złożoność przeszukiwania macierzy sąsiedztwa jest mniejsza.

5 Bibliografia

Literatura

- [1] S. Prata: Język C++ Szkoła programowania wyd. VI (2013)
- [2] W. Anggoro: C++ Struktury danych i algorytmy (2019)
- [3] https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda
- [4] A. Drozdek: C++. Algorytmy i Struktury Danych