Projektowanie Algorytmów i Metody Sztucznej Inteligencji
Projekt 2

Prowadzący: Marta Emirsajłow

Maksymilian Kadukowski
248974

13 Maj 2020

1 Wstęp

1.1 Cel ćwiczenia

Poznanie dwóch głównych reprezentacji grafu w informatyce oraz przetestowanie ich wydajności w zależności od algorytmu.

2 Teoria

W sprawozdaniu pojawiają się następujące skróty:

- V - liczba wierzchołków,
- E - liczba krawędzi.

Gęstość grafu zdefiniowan jest jako $D = \frac{E}{V(V-1)}$.

2.1 Reprezentacje grafu

W ćwiczeniu badania zostały poddane następujące reprezentacje grafu:

1. macierzowa

   Reprezentacja macierzowa polega na stworzeniu macierzy rozmiarach VxV. W ten sposób dostęp do informacji o krawędzi między dwoma wierzchołkami uzyskujemy w czasie stałym (O(1)), co jest główną zaletą takiej reprezentacji.

2. listowa

   Reprezentacja listowa polega na zapisywaniu informacji o krawędziach dla każdego wierzchołka. Zaletą takiego rozwiązania jest stały czas dostępu do wszystkich krawędzi wychodzących/wchodzących do danego wierzchołka.

2.2 Algorytm Dijkstry

Algorytm opracowany przez Edsger'a W. Dijkstrę w 1956, służący do znajdywania najkrótszej drogi pomiędzy wierzchołkami grafu. Oryginalnie znajdował najkrótszą drogę pomiędzy dwoma wierzchołkami, obecnie jego najpopularniejsza wersja znajduje najkrótszą drogę do wszystkich wierzchołków w grafie z zadanego na początku wierzchołka.

2.3 Przewidywane wyniki

W ćwiczeniu badane są: algorytm dla reprezentacji macierzowej o czasie $O(V^2)$ oraz algorytm dla reprezentacji listowej o czasie $O(E \log V)$.

Na ogół algorytm dla reprezentacji listowej powinien być szybszy, szczególnie dla grafu o niskiej gęstości.

3 Program i sprzęt

Program w celu wykonania ćwiczenia posiada proste menu w linii komend, pozwalające na wykonanie testów, wczytanie pliku zawierającego graf, zapis wyniku działania algorytmu Dijkstry oraz dodatkowo zapis wyniku w języku GraphViz, dła łatwiejszego debugowania.

Testy przeprowadzone były na procesorze i5-3320M.

4 Wyniki

4.1 Tabele

W tabelach przedstawiony został uśredniony wynik ze 100 testów dla danej reprezentacji. Pełny graf definiuje jako gęstość równą 100%.

- Reprezentacja macierzowa

V	D	t [ms]
10	25%	0.0024
10	50%	0.0034
10	75%	0.0026
10	100%	0.0020
50	25%	0.0242
50	50%	0.0219
50	75%	0.0166
50	100%	0.0144
100	25%	0.0595
100	50%	0.0732
100	75%	0.0629
100	100%	0.0460
500	25%	1.5334
500	50%	2.0459
500	75%	2.6247
500	100%	1.4530
1000	25%	6.6175
1000	50%	8.7684
1000	75%	10.7416
1000	100%	5.1181

- Reprezentacja listowa

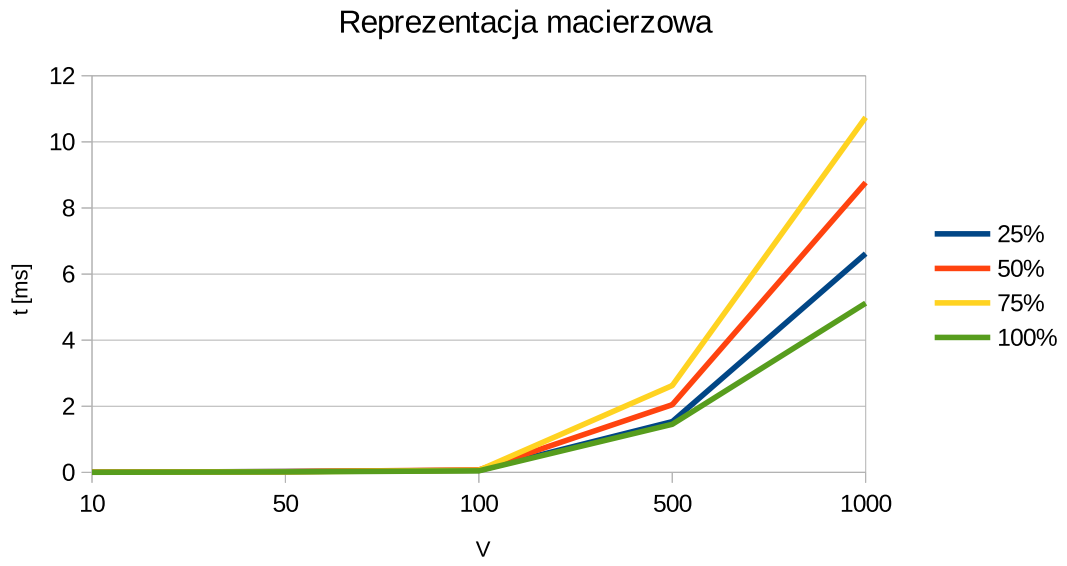
V	D	t [ms]
10	25%	0.0006
10	50%	0.0008
10	75%	0.0009
10	100%	0.0009
50	25%	0.0058
50	50%	0.0083
50	75%	0.0111
50	100%	0.0097
100	25%	0.0158
100	50%	0.0229
100	75%	0.0282
100	100%	0.0360
500	25%	0.3756
500	50%	1.0803
500	75%	1.7285
500	100%	0.6005
1000	25%	2.5752
1000	50%	5.8309
1000	75%	9.3620
1000	100%	2.1618

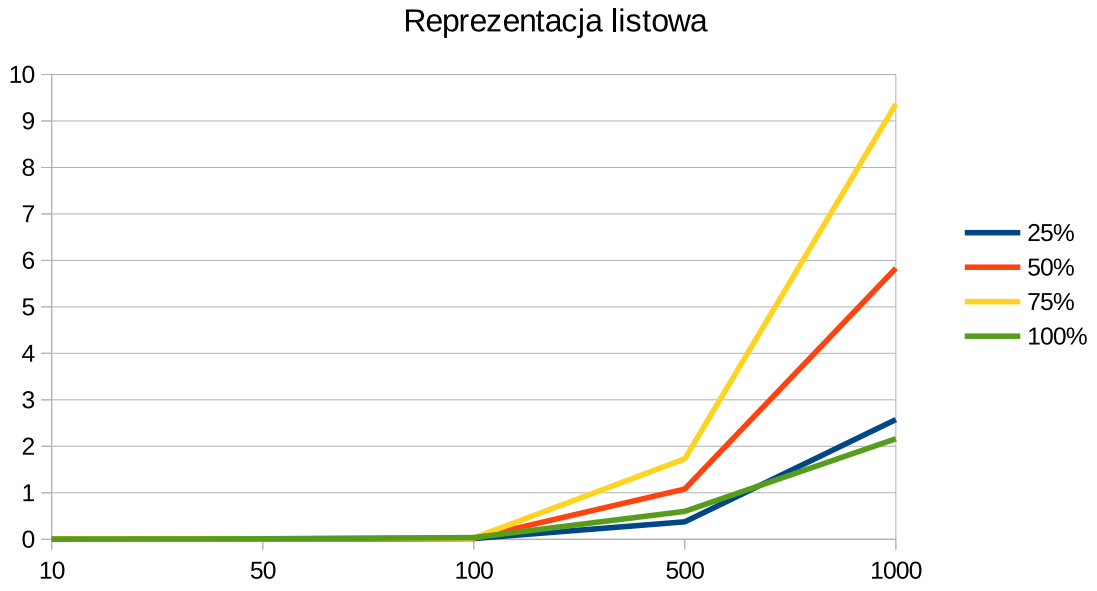
Zgodnie z przewidywaniami reprezentacja listowa jest szybsza, szczególnie dla małej gęstości grafu. W dużych gęstościach różnica już nie jest tak duża, powodem jest zbliżanie się liczby E do liczby V^2.

Ciekawą anomalią jest zmniejszenie się czasu potrzebnego do wykonania algorytmu dla grafu pełnego. Sytuacja występuje dla 2 reprezentacji grafu oraz dla losowych wag o dystrybucji normalnej i jednolitej (dodatkowe testy znajdują się na branchy `boost-validators`).

4.2 Wykresy

W zależności od reprezentacji i dla każedj gęstości:

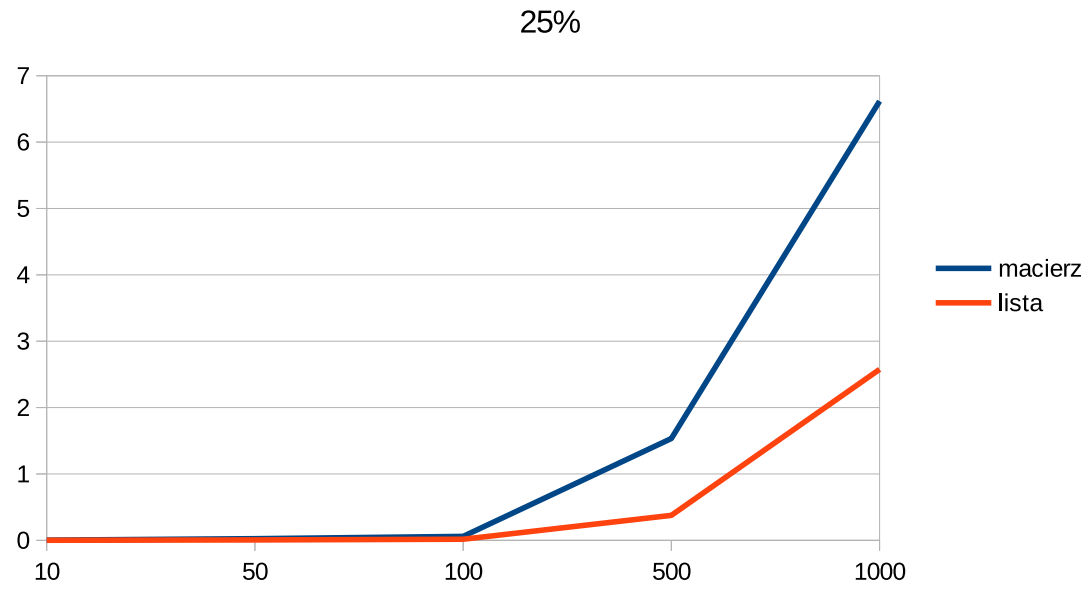


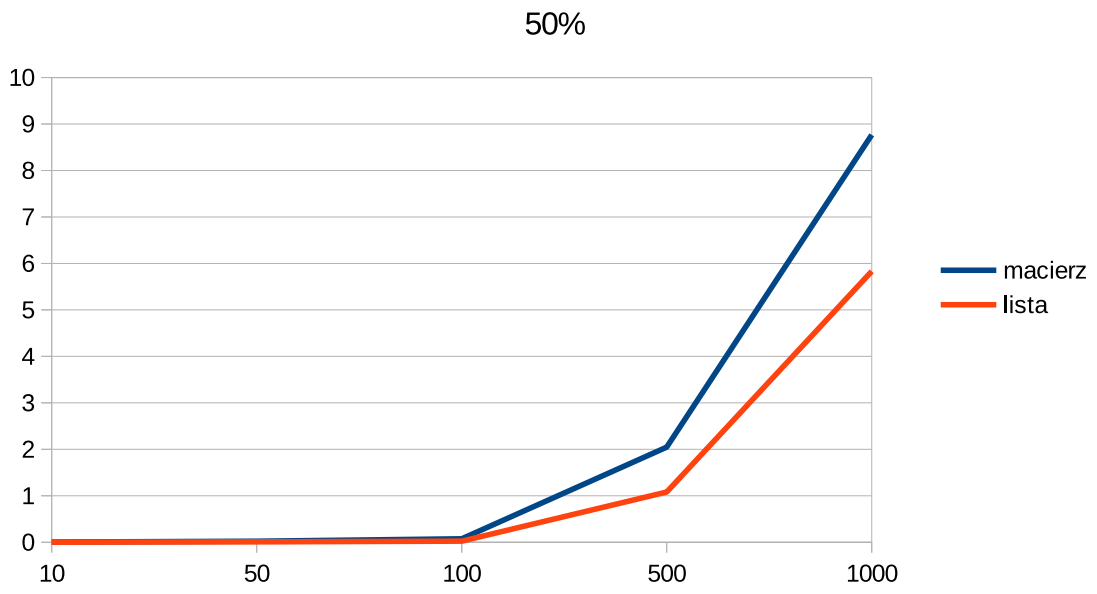


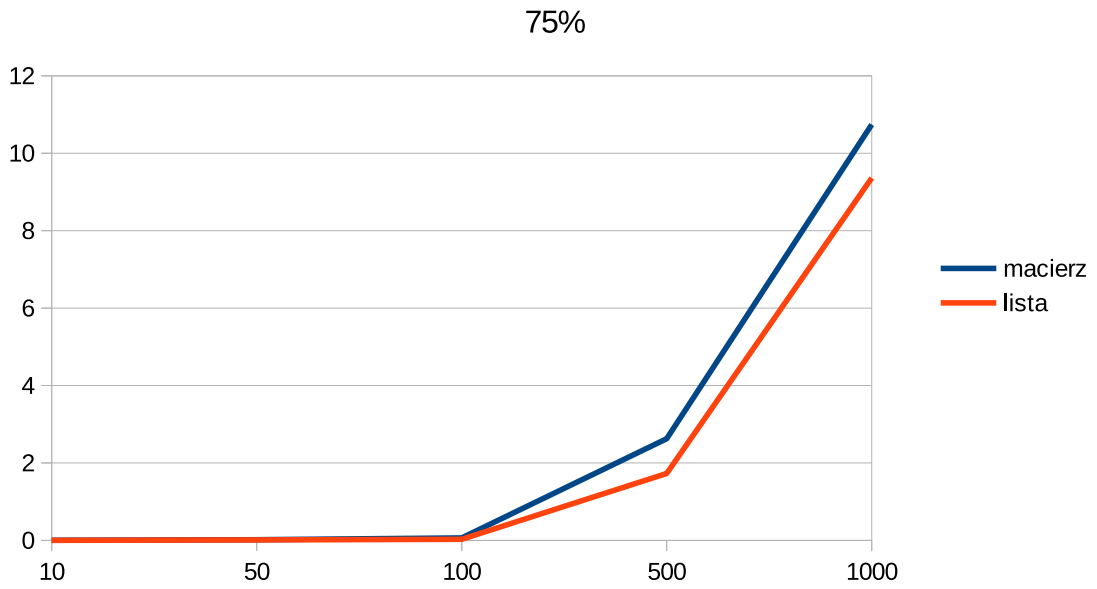
Mimo braku zależności algorytmu macierzowego od ilości krawędzi widać lekki wzrost czasu dla większej gęstości.

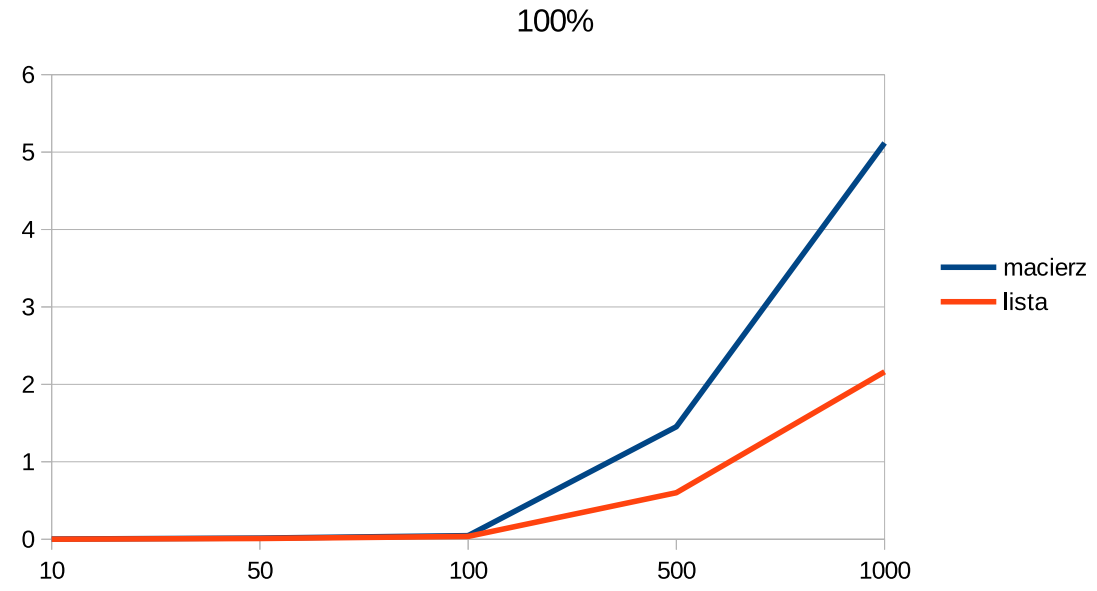
W przypadku reprezentacji listowej zależność od liczby krawędzi jest już o wiele bardziej widoczna, co zgadza się z przewidywaniami.

W zależności od gęstości grafu:


25%
7
6
5
4
3
2
1
0
10
50
100
500
1000
macierz
lista


50%
10
9
8
7
6
5
4
3
2
1
0
10
50
100
500
1000
macierz
lista





Jak w przewidywaniach, reprezentacja listowa jest szybsza od macierzowej, szczególnie dla małej gęstości grafu.

5 Podsumowanie i wnioski

Czas działania algorytmu dla reprezentacji listowej

Niestety czas działania macierzy nie jest niezależny od gęstości, co niejako kłóci się z założeniem o wydajności algorytmu równym $O(V^2)$. Przyczyną może być nie optymalnie zaimplementowany algorytm.

Zgodnie z przewidywaniami czas potrzebny do wykonania algorytmu dla listy wzrasta wraz z ilością krawędzi (za wyjątkiem grafu pełnego). Nie jestem w stanie wytłumaczyć dlaczego taka sytuacja zaistniała.

Zgodnie z przewidywaniami algorytm działa szybciej dla reprezentacji listowej niż dla macierzy, szczególnie dla małych gęstości.