Struktury danych i złożoność obliczeniowa
projekt 2

Porowski Wiktor 252719

June 2021

Spis treści

1 Wstęp

Celem projektu jest zaimplementowanie oraz dokonanie pomiaru czasu działania wybranych algorytmów grafowych rozwiązujących następujące problemy:
a) wyznaczanie minimalnego drzewa rozpinającego (MST) algorytm Prima
b) wyznaczanie najkrótszej ścieżki w grafie – algorytm Dijkstry
Algorytmy te zaimplementowane są dla obu poniższych reprezentacji grafu w pamięci komputera:
- reprezentacja macierzowa (macierz wag - macierz sąsiedztwa w której zamiast wartości 0/1 wpisane są wagi krawędzi, umowna wartość oznacza brak krawędzi)
- reprezentacja listowa (połączone listy sąsiadów).

2 Plan eksperymentu i założenia

Wszystkie struktury danych są alokowane dynamicznie
• waga (przepustowość) krawędzi jest liczbą całkowitą
• pomiary czasu działania algorytmów wykonywane są w zależności od rozmiaru grafu oraz jego gęstości (stosunku liczby krawędzi do maksymalnej możliwej liczby krawędzi przy danej liczbie wierzchołków).
Badania wykonane dla 5 różnych (reprezentatywnych) liczb wierzchołków oraz następujących gęstości grafu: 25%, 50%, 75% oraz 99%. Dla każdego zestawu: reprezentacja grafu, liczba wierzchołków i gęstość wygenerowane jest po 100 losowych instancji.

3 Sposób generowania grafu

Na początku by zagwarantować spójność generowana jest tablica o rozmiarze takim jaka jest liczba wierzchołków grafu gdzie wartością elementu tabicy jest indeks tablicy. Tablice mieszamy. Następnie iterujemy po niej i tworzymy połączenia w grafie odpowiadojące sąsiednim wartością w tablicy.
Teraz spójność jest już zagwarantowana. Pozostałe krawędzie generujemy w losowy sposób sprawdzając czy krawędz którą wylosowaliśmy nie istnieje już w grafie.

4 Oczekiwane złożoności czasowe wykonywania operacji

Algorytm Dijkstry
Z wykorzystaniem naiwnego wyszukiwania liniowego (zamaist zastosowania kopca): Dla list sąsiedzwa złożoność czasowa to $0(E + V^2)$ Ponieważ V razy znajdujemy min w tablicy ($O(V^2)$) i dokładnie raz każda krawędź z listy sąsiedztwa jest sprawdzana (O(E)) Graf jest spójny wieć $E > V - 2$. Więc złożoność czasowa w zasadzie to $0(V^2)$ W przypadku macierzy wag V razy znajdujemy min w tablicy ($O(V^2)$) oraz zawsze sprawdzamy krawędzie dla wierzchołka V razy w pętli głównej która wykonuje się V razy. Więc złóżoność to również $0(V^2)$

Algorytm Prima
Z wykorzystaniem kopca: Pętla główna wykonuje się V razy w której dodajemy oraz usuwamy krawędzie z kopca. Musimy zauważyć ze dodamy/usuniemy conajwyżej wszystkie krawędzie (E), kopiec może mieć maksymalnie rozmiar E. Czas operacji na kopcu to O(logE). tak więc złożoność czasowa to O(E*logE)

5 Zestawienie wyników pomiarów

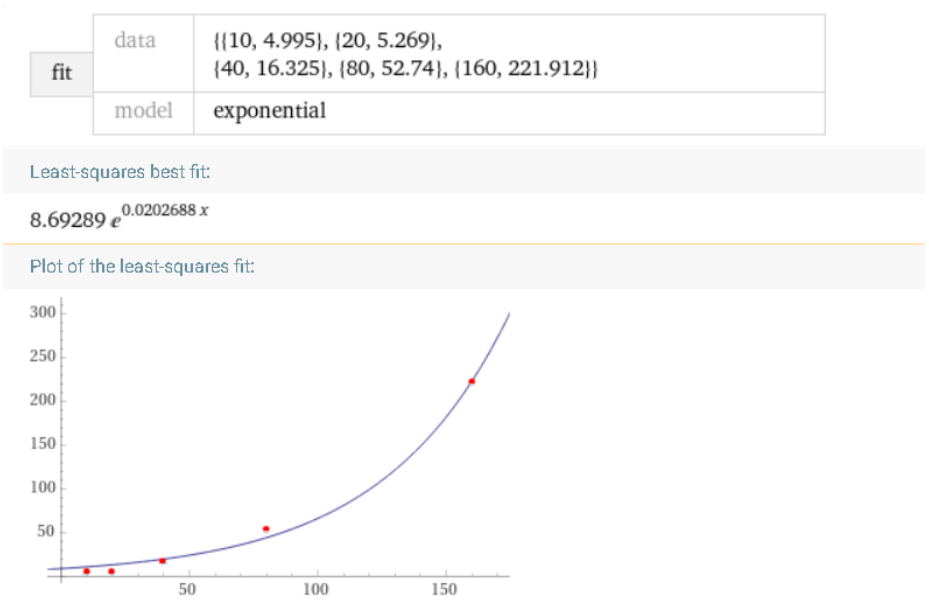
```
SIZE OF STRUCTURE = 10 DENSITY = 25
Time [us] = 4.995 -dijikstra_l
Time [us] = 6.232 -dijikstra_m
Time [us] = 24.936 -prim_l
Time [us] = 27.204 -prim_m

SIZE OF STRUCTURE = 10 DENSITY = 50
Time [us] = 4.313 -dijikstra_l
Time [us] = 6.62 -dijikstra_m
Time [us] = 49.473 -prim_l
Time [us] = 52.192 -prim_m

SIZE OF STRUCTURE = 10 DENSITY = 75
Time [us] = 2.749 -dijikstra_l
Time [us] = 3.527 -dijikstra_m
Time [us] = 43.63 -prim_l
Time [us] = 41.445 -prim_m

SIZE OF STRUCTURE = 10 DENSITY = 99
Time [us] = 3.057 -dijikstra_l
Time [us] = 4.38 -dijikstra_m
Time [us] = 57.318 -prim_l
Time [us] = 57.178 -prim_m

SIZE OF STRUCTURE = 20 DENSITY = 25
Time [us] = 5.269 -dijikstra_l
Time [us] = 7.692 -dijikstra_m
Time [us] = 75.331 -prim_l
Time [us] = 78.263 -prim_m

SIZE OF STRUCTURE = 20 DENSITY = 50
Time [us] = 6.301 -dijikstra_l
Time [us] = 11.372 -dijikstra_m
Time [us] = 153.718 -prim_l
Time [us] = 157.035 -prim_m

SIZE OF STRUCTURE = 20 DENSITY = 75
Time [us] = 7.573 -dijikstra_l
Time [us] = 10.408 -dijikstra_m
Time [us] = 234.809 -prim_l
Time [us] = 225.872 -prim_m

SIZE OF STRUCTURE = 20 DENSITY = 99
Time [us] = 8.599 -dijikstra_l
Time [us] = 10.05 -dijikstra_m
Time [us] = 312.364 -prim_l
Time [us] = 291.722 -prim_m
```

```
SIZE OF STRUCTURE = 40 DENSITY = 25
Time [us] = 16.325 -dijikstra_l
Time [us] = 26.86 -dijikstra_m
Time [us] = 420.804 -prim_l
Time [us] = 423.038 -prim_m

SIZE OF STRUCTURE = 40 DENSITY = 50
Time [us] = 20.814 -dijikstra_l
Time [us] = 34.137 -dijikstra_m
Time [us] = 940.872 -prim_l
Time [us] = 932.111 -prim_m

SIZE OF STRUCTURE = 40 DENSITY = 75
Time [us] = 24.138 -dijikstra_l
Time [us] = 29.538 -dijikstra_m
Time [us] = 1402.89 -prim_l
Time [us] = 1379.57 -prim_m

SIZE OF STRUCTURE = 40 DENSITY = 99
Time [us] = 28.054 -dijikstra_l
Time [us] = 23.42 -dijikstra_m
Time [us] = 2017.92 -prim_l
Time [us] = 1962.62 -prim_m

SIZE OF STRUCTURE = 80 DENSITY = 25
Time [us] = 52.74 -dijikstra_l
Time [us] = 88.077 -dijikstra_m
Time [us] = 2721.49 -prim_l
Time [us] = 2827.41 -prim_m

SIZE OF STRUCTURE = 80 DENSITY = 50
Time [us] = 82.82 -dijikstra_l
Time [us] = 118.164 -dijikstra_m
Time [us] = 7410.34 -prim_l
Time [us] = 7338.86 -prim_m

SIZE OF STRUCTURE = 80 DENSITY = 75
Time [us] = 147.943 -dijikstra_l
Time [us] = 109.347 -dijikstra_m
Time [us] = 13987.4 -prim_l
Time [us] = 14076 -prim_m

SIZE OF STRUCTURE = 80 DENSITY = 99
Time [us] = 195.465 -dijikstra_l
Time [us] = 78.055 -dijikstra_m
Time [us] = 22462.1 -prim_l
Time [us] = 22792.1 -prim_m
```

```
SIZE OF STRUCTURE = 160 DENSITY = 25
Time [us] = 221.912 -dijikstra_l
Time [us] = 312.776 -dijikstra_m
Time [us] = 27021.7 -prim_l
Time [us] = 26601.2 -prim_m

SIZE OF STRUCTURE = 160 DENSITY = 50
Time [us] = 484.181 -dijikstra_l
Time [us] = 424.938 -dijikstra_m
Time [us] = 92128.3 -prim_l
Time [us] = 93336.5 -prim_m

SIZE OF STRUCTURE = 160 DENSITY = 75
Time [us] = 896.43 -dijikstra_l
Time [us] = 389.731 -dijikstra_m
Time [us] = 193772 -prim_l
Time [us] = 192654 -prim_m

SIZE OF STRUCTURE = 160 DENSITY = 99
Time [us] = 1231.73 -dijikstra_l
Time [us] = 279.817 -dijikstra_m
Time [us] = 343286 -prim_l
Time [us] = 341916 -prim_m
```

5.1 Dijkstra lista sąsiedztwa

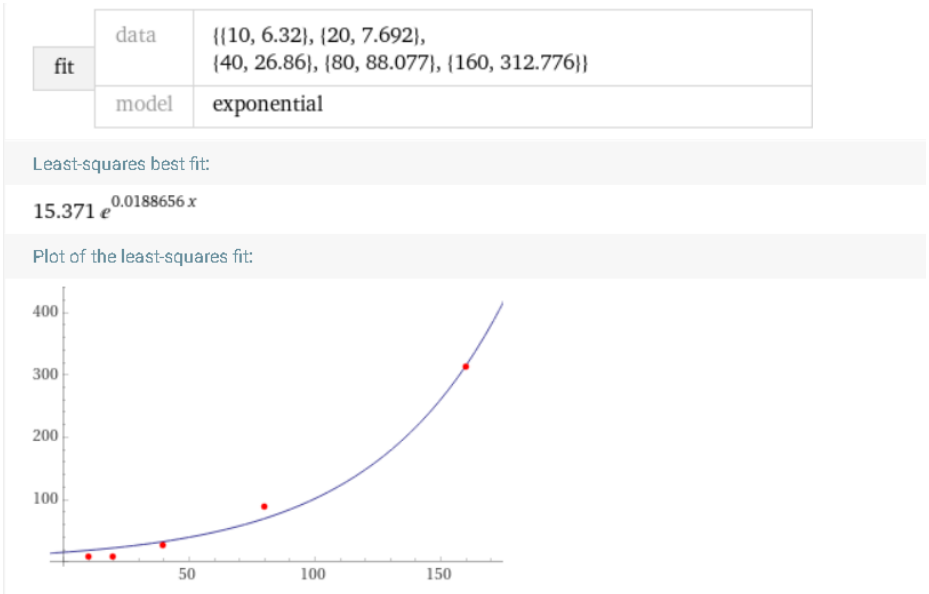
Widać że gęstość grafu w niewielkim stopniu ma wpływa na czas wykonania. Wynika to z większego E w $0(E + V^2)$.
Istotną różnice widać przy zwiększeniu ilości wierzchołów wyniki potwierdzają przewidywnia: $0(V^2)$



Rysunek 1: density 25%

5.2 Dijkstra macierz wag

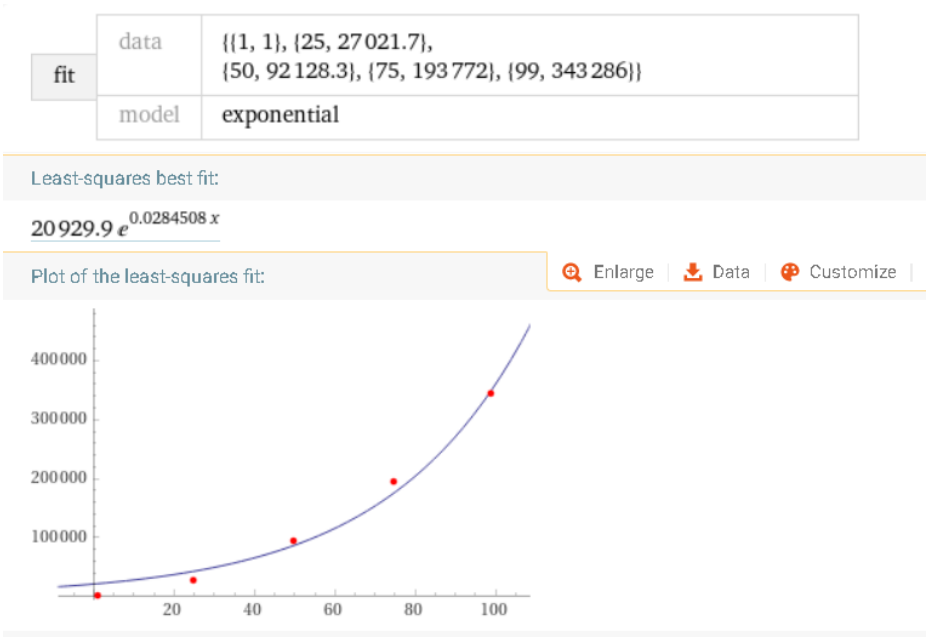
Widać że gęstość grafu nie wpływa na czas wykonania.
Istotną różnicy widać przy zwiększeniu ilości wierzchołów wyniki potwierdzają przewidywnia: $0(V^2)$



Rysunek 2: density 25%

5.3 Prim lista sąsiedztwa

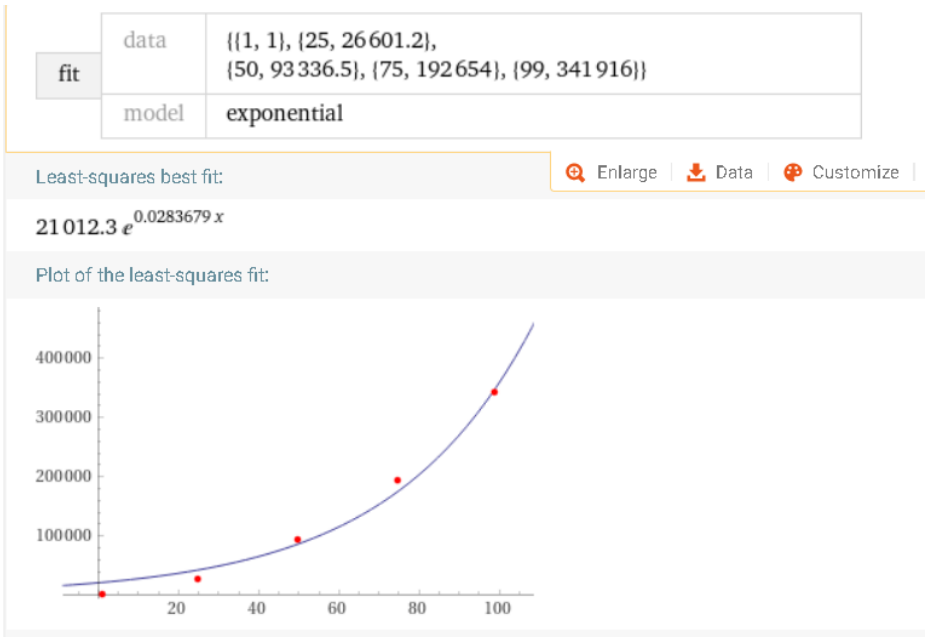
Czasy rosną szybciej od oczekiwanych powedem tego jest fakt że gwarantujemy aby kopiec cały czas zajmował minimalną ilosć pamięci więc każda operacja wstawiania/usuwania wiąże się ze zmianą rozmiaru tablicy w której jest zaimplementownany kopiec , tak więc operacje na kopcu zamiast O(logE) zajmują czas O(E) co prowadzi ze złożoności O(E*logE) do $O(E^2)$. Łatwo to zaobserwować dla konkretnego rozmiaru grafu zmnieniając jego gęstość.



Rysunek 3: size 160

5.4 Prim macierz wag

Te same uwagi co dla listy sąsiedztwa



Rysunek 4: size 160

6 Wnioski

Reprezentacja listowa bardziej optymalna pamięciowo w dodatku posiadała dodatkowy plus w postaci odrobine mniejszych nakładów obliczeniowych w przypadku sprawdzania sąsiadów (sprawdzamy sąsiadów danego wierczhołka tylko tych którzy istnieją max V-1) dla macierzy wag zawsze sprawdzamy V-1 razy nawet jak wierchołek ma mniej sąsiadów.
Aby uniknąć pogorszenia złożoności czasowej w przypadku wyznaczania minimalnego drzewa rozpinającego powinniśmy zrezygnować z tego żeby kopiec zajmował minimalną ilość miejsca i ustalić jego wielkość na E.