Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 2

*Imię i nazwisko:* Jakub Zając

*Nr indeksu:* 259362

*Dane prowadzącego:* Marta Emirsajłow

*Termin zajęć:* Poniedziałek 15:15

1. Wprowadzanie

Zadanie projektowe polegało na przeprowadzeniu analizy efektywności wybranego algorytmu znajdowania najkrótszej ścieżki w grafie. Należało go przetestować dla grafu reprezentowanego jako lista i macierz sąsiedztwa.

Algorytm, który wybrałem to algorytm*Bellmana-Forda*. Został on przetestowany dla następujących parametrów:

1. Liczba wierzchołków grafów:

- 10 000

- 50 000

- 100 000

- 500 000

- 1 000 000

1. Wypełnienie grafów:

- tablica całkowicie nieposortowana

- 25%

- 50%

- 75%

- 100%

1. Reprezentacja grafów:

- macierz sąsiedztwa

- lista sąsiedztwa

1. Opis reprezentacji grafów
2. Macierz sąsiedztwa – reprezentacja grafu poprzez macierz, której indeksy oznaczają numery wierzchołków, a elementy wartości wag krawędzi.
3. Lista sąsiedztwa – reprezentacja grafu opierająca się na tabeli list reprezentującej wierzchołki, której elementy to inne wierzchołki z nimi połączone z polem zawierającym wagę krawędzi między nimi.
4. Opisy algorytmu Bellmana-Forda

Jest to algorytm służący do znajdywania najkrótszych ścieżek w grafie ważonym z dowolnie wybranego wierzchołka do pozostałych wierzchołków grafu. Algorytm można wykorzystać w grafach skierowanych oraz z ujemnymi wagami krawędzi. Opiera się na zasadzie relaksacji, występującej o raz mniej niż liczba wierzchołków dla każdej krawędzi grafu.

Złożoność obliczeniowa dla macierzy sąsiedztwa wynosi O(|V|^3) – liczba wierzchołków do potęgi 3. Złożoność ta wynika z V iteracji przez wszystkie krawędzie, których przez reprezentację grafu jest |V|^2. Dla listy sąsiedztwa złożoność wynosi O(|V|\*|E|). Dla grafu pełnego liczba krawędzi |E| = |V|^2 co skutkuje taką samą złożonością jak dla macierzy, ale dla mniejszych gęstości (<100%) dla listy algorytm zawsze wykona się szybciej.

1. Przebieg eksperymentów i uzyskane rezultaty
2. Czasy dla macierzy sąsiedztwa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MACIERZ | 25% | 50% | 75% | 100% |
| 50 | 0.33034 | 0.34013 | 0.35007 | 0.37085 |
| 100 | 2.64969 | 2.63523 | 2.65818 | 2.66334 |
| 150 | 8.7745 | 8.70883 | 8.73821 | 8.80745 |
| 200 | 20.5047 | 20.5331 | 20.5256 | 20.4772 |
| 250 | 39.9244 | 39.9303 | 40.0755 | 40.1003 |

1. Czasy dla listy sąsiedztwa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LISTA | 25% | 50% | 75% | 100% |
| 50 | 0.07001 | 0.12014 | 0.29006 | 0.37007 |
| 100 | 0.77026 | 1.4544 | 2.12516 | 2.87049 |
| 150 | 2.417 | 4.56696 | 7.16929 | 8.86117 |
| 200 | 3.81135 | 7.63771 | 15.8617 | 19.0581 |
| 250 | 7.974 | 23.092 | 32.1338 | 40.976 |

1. Podsumowanie i wnioski

Wyniki eksperymentów potwierdziły oczekiwania. Lista sąsiedztwa dla większości przypadków cechuje się lepszą wydajnością dla algorytmu szukania najkrótszej ścieżki Bellmana-Forda. Jedynym przypadkiem równoważnym złożeniowo jest graf pełny. Dodatkowo zauważyć możemy eksponencjalny wzrost czasu trwania algorytmu razem ze zwiększaniem liczby wierzchołków.

Mimo trudniejszej implementacji warto jest wykorzystać listę do rozwiązania problemu najkrótszej ścieżki.

1. Literatura
2. Thomas H. Cormen – Introduction to Algorithms, 3rd Edidtion
3. https://en.wikipedia.org/wiki/Bellman-Ford\_algorithm
4. Wykłady i prezentacje dr inż. Łukasza Jelenia