PROJEKTOWANIE ALGORYTMÓW I METODY SZTUCZNEJ INTELIGENCJI

Marcin Radke

03.05.20

Piątek 7:30

Mgr Marta Emirsajłow

Projekt nr 2

1. Opis

Celem zadania było napisanie programu, który miał za zadanie znalezienie najkrótszej drogi w grafie wczytanego z pliku korzystając z algorytmu Bellmana-Forda.

Badania zostały wykonane dla 5 różnych wielkości wierzchołków:

* 50
* 100
* 150
* 200
* 250

oraz dla gęstości grafu:

* 25%
* 50%
* 75%
* 100%

1. Algorytm

Algorytm Bellmana-Forda służy do wyszukiwania najkrótszych ścieżek w grafie ważonym z wierzchołka źródłowego do wszystkich pozostałych wierzchołków w grafie. Idea algorytmu opiera się na metodzie relaksacji. W odróżnieniu od algorytmu Dijkstry, algorytm Bellmana-Forda działa poprawnie także dla grafów z wagami ujemnymi (nie może jednak wystąpić cykl o łącznej ujemnej wadze osiągalny ze źródła).

Złożoność czasowa dla implementacji:

* Macierz:
* Lista:

W – wierzchołki, K – krawędzie

1. Wyniki testów
   1. Graf w jako lista:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lista | | Gęstość | | | |
| 0,25 | 0,5 | 0,75 | 1 |
| Liczba  elementów | 50 | 0,496 | 0,496 | 0,497 | 0,994 |
| 100 | 1,488 | 2,48 | 4,47 | 5,95 |
| 150 | 4,96 | 9,992 | 15,375 | 21,33 |
| 200 | 12,4 | 26,293 | 37,699 | 52,569 |
| 250 | 24,826 | 50,07 | 74,374 | 103,1 |

* 1. Graf jako macierz:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Macierz | | Gęstość | | | |
| 0,25 | 0,5 | 0,75 | 1 |
| Liczba  elementów | 50 | 0,493 | 0,494 | 0,496 | 0,497 |
| 100 | 0,991 | 1,948 | 2,974 | 3,498 |
| 150 | 2,976 | 6,421 | 9,92 | 13,365 |
| 200 | 8,402 | 15,898 | 24,304 | 31,709 |
| 250 | 14,878 | 31,717 | 45,632 | 63,461 |

* 1. Gęstość grafu:
  2. 25%
  3. 50%
  4. 75%
  5. 100%

1. Wnioski

* na podstawie wyników można zobaczyć, że metoda listowa jest coraz wolniejsza ze względu na złożoność obliczeniową
* metoda listowa jest lepsza dla grafów o małej ilości krawędzi
* dla małej ilości wierzchołków obie metody mają podobne czasy szukania rozwiązania

1. Bibliografia

* <https://pl.wikipedia.org/wiki/Algorytm_Bellmana-Forda>
* <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/>
* <https://www.geeksforgeeks.org/bellman-ford-algorithm-simple-implementation/>