**Raport**

**Projekt 2**

**Projektowanie algorytmów i metody sztucznej inteligencji**

**Wykonał:** Michał Burda 241484  
**Data oddania:** 15.05.2019  
**Termin zajęć:** śr. 18:55

**Temat projektu:** Grafy – algorytm Dijkstry.

**1. Wstęp** Jako zadanie należało zaimplementować graf w postaci listy sąsiedztwa i macierzy sąsiedztwa oraz algorytm Dijkstry pozwalający na znalezienie najkrótszej drogi między wierzchołkami grafu. Dla każdego sposobu reprezentacji stworzono grafy o gęstości 25%, 50%, 75% i graf pełny. Dla każdej gęstości grafu stworzono grafy o ilości wierzchołków: 50, 100, 200, 300, 500. Dla każdej z kombinacji program wykonał 100 powtórzeń pomiaru czasu wykonywania algorytmu Dijkstry dla losowego wierzchołka w celu uśrednienia wyników.  
 Pesymistyczna złożoność obliczeniowa algorytmu Dijkstry wynosi O(E+VlogV).

**2. Wyniki pomiarów**

Tabela - Średni czas wykonywania pojedynczego algorytmu w milisekundach – macierz sąsiedztwa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gęstość**  **Ilość grafu**  **wierzchołków** | 25% | 50% | 75% | pełny |
| 50 | 0,003 | 0,006 | 0,008 | 0,004 |
| 100 | 0,006 | 0,013 | 0,016 | 0,008 |
| 200 | 0,035 | 0,054 | 0,052 | 0,047 |
| 300 | 0,087 | 0,147 | 0,146 | 0,107 |
| 500 | 0,286 | 0,455 | 0,402 | 0,340 |

Rysunek - zależność czasu wykonywania od gęstości grafu i ilości wierzchołków - macierz sąsiedztwa

Tabela - Średni czas wykonywania pojedynczego algorytmu w milisekundach - lista sąsiedztwa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gęstość**  **Ilość grafu**  **wierzchołków** | 25% | 50% | 75% | pełny |
| 50 | 0,003 | 0,004 | 0,006 | 0,008 |
| 100 | 0,007 | 0,010 | 0,016 | 0,021 |
| 200 | 0,035 | 0,049 | 0,059 | 0,081 |
| 300 | 0,059 | 0,095 | 0,145 | 0,188 |
| 500 | 0,142 | 0,272 | 0,378 | 0,540 |

Rysunek - zależność czasu wykonywania od gęstości grafu i ilości wierzchołków – lista sąsiedztwa

**3. Podsumowanie** Wyniki pomiarów różnią się w zależności od sposobu reprezentacji grafu głównie pod kątem zależności od gęstości grafu. W przypadku macierzy sąsiedztwa, algorytm najszybciej wykonał się dla gęstości 0,25, następnie dla grafu pełnego, dla gęstości 0,75 i najwolniej dla gęstości 0,5. W przypadku listy sąsiedztwa, algorytm najszybciej wykonał się dla gęstości 0,25, następnie dla gęstości 0,5, dla gęstości 0,5 i najwolniej dla dla grafu pełnego. Różnice te mogą być spowodowane implementacją w programie i mogłyby być zupełnie różne dla innych implementacji. Sam algorytm wydaje się działać poprawnie.