|  |  |
| --- | --- |
| **POLITECHNIKA WROCŁAWSKA**  **Wydział: Informatyka techniczna** | Roman Yavorenko 276091 |
| **Algorytmy i złożoność obliczeniowa** | |
| BADANIE EFEKTYWNOŚCI ALGORYTMÓW GRAFOWYCH W ZALEŻNOŚCI OD ROZMIARU INSTANCJI I SPOSOBU PAMIĘTANIA GRAFU | |
|

# Wstęp:

Implementacja algorytmów grafowych oraz analiza ich efektywności to kluczowy element informatyki teoretycznej i praktycznej. Projekt skupia się na porównaniu dwóch głównych reprezentacji grafów — macierzowej i listowej — pod kątem złożoności czasowej i przestrzennej algorytmu Dijkstry, używanego do wyszukiwania najkrótszych ścieżek w grafie.

Algorytm Dijkstry jest powszechnie stosowany w różnych dziedzinach informatyki i technologii, gdzie kluczową rolę odgrywa efektywność operacji na grafach. Wybór odpowiedniej reprezentacji grafu ma istotny wpływ na szybkość działania algorytmu oraz zużycie pamięci.

Dla każdej wielkości grafu zostaną wygenerowane grafy o różnej gęstości 25%, 50%, 75%

Pomiary były wykonane dla takiej ilości wierzchołków: 1000, 5000,10000

Spszęt:

Procesor: AMD Ryzen 5 3550H

Operacyjna pamięć: 16Gb

Dysk: SSD 500Gb

Implementacja tych algorytmów została zrealizowana w języku C++.

**Tablicy dla różnej ilości wierszchołków**

Czas dla 1000 wierzchołków

Зображення, що містить текст, знімок екрана, ряд, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, число, схема

Автоматично згенерований опис

Czas dla 5000 wierzchołków

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, число, схема

Автоматично згенерований опис

Czas dla 10000 wierzchołków

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, число, Графік

Автоматично згенерований опис

# Wnioski:

Na podstawie przeprowadzonych badań porównawczych efektywności reprezentacji grafów (macierzowej i listowej) dla algorytmu Dijkstry, można wysunąć następujące wnioski:

1. **Czas wykonania algorytmu w zależności od liczby wierzchołków:**
   * Dla reprezentacji listowej: Zwiększenie liczby wierzchołków skutkuje proporcjonalnym wzrostem czasu wykonania. Na przykład dla 1000 wierzchołków czas wykonania waha się od 6ms do 10ms w zależności od gęstości grafu.
   * Dla reprezentacji macierzowej: Czas wykonania również rośnie wraz ze wzrostem liczby wierzchołków, jednak zauważalny jest większy wzrost czasu w porównaniu do reprezentacji listowej. Na przykład dla 1000 wierzchołków czas wykonania wynosi od 8ms do 10ms.
2. **Porównanie efektywności reprezentacji grafów:**
   * Reprezentacja listowa często cechuje się szybszym czasem dostępu do sąsiadów danego wierzchołka, co może być korzystne w algorytmach wymagających częstych operacji na sąsiedztwie wierzchołków, jak algorytm Dijkstry.
   * Reprezentacja macierzowa, mimo większego zużycia pamięci, może być szybsza w operacjach sprawdzania istnienia krawędzi między wierzchołkami.
3. **Wpływ gęstości grafu na czas wykonania:**
   * W badaniach zauważono, że gęstość grafu ma istotny wpływ na czas wykonania algorytmu Dijkstry. Im większa gęstość, tym zazwyczaj dłuższy czas obliczeń, co wynika z większej liczby krawędzi do przetworzenia przez algorytm.
4. **Rozbieżności pomiędzy teoretycznymi a praktycznymi wynikami:**
   * W praktyce czas wykonania algorytmu może różnić się od teoretycznych złożoności obliczeniowych ze względu na dodatkowe czynniki, takie jak implementacyjne szczegóły języka programowania, sposób alokacji pamięci, czy charakterystyka testowanego sprzętu.

Podsumowując, reprezentacja grafu oraz jego gęstość mają istotny wpływ na wydajność algorytmu Dijkstry. Wybór odpowiedniej reprezentacji powinien być dostosowany do konkretnych potrzeb aplikacji, biorąc pod uwagę zarówno czas wykonania, jak i zużycie pamięci. Dalsze badania mogą skupić się na optymalizacji implementacji algorytmu Dijkstry w kontekście różnych reprezentacji grafu oraz bardziej szczegółowej analizie wpływu różnych czynników na czas działania algorytmów grafowych.