

Sprawozdanie PAMSI

Wyszukiwanie najkrótszej ścieżki w grafie nieskierowanym.

- **Wprowadzenie**

Graf jest bardzo dobrą strukturą do zapisywania zależności pomiędzy np. fizycznymi obiektami. Jednym z przykładów, może być mapa. Wierzchołki grafu odpowiadają miastom umieszczonym na mapie, a krawędzie pomiędzy wierzchołkami odpowiadają drogom łączącym miasta. Dodatkowo krawędzie posiadają swoje „wagi”, co może odpowiadać fizycznym odległościom pomiędzy miastami. Mając rozumianą w ten sposób strukturę, brakuje nam tylko algorytmu dzięki, któremu będziemy mogli poznać najkrótszą drogę pomiędzy miastem startowym a końcowym.

Jednym z algorytmów pozwalających wyłonić najkrótszą drogę pomiędzy (w tym przypadku) miastami, jest algorytm Branch and Bound. Aby zwiększyć wydajność pracy algorytmu, przy jego implementacji wykorzystałem tablicę odwiedzin (*visited_table[graph_size]*). Po znalezieniu ścieżki o najmniejszej wadze, prowadzącej do wierzchołka, odznaczam wierzchołek jako „sprawdzony” i pomijam go w kolejnych poszukiwaniach. Takie działanie znacznie skraca czas pracy algorytmu, ponieważ nie wykonujemy niepotrzebnych ruchów aby sprawdzać inne ścieżki (zawsze dłuższe) prowadzące do odnanzonego wierzchołka. Przy implementacji użyłem też tablic *final_costs[graph_size]*, przechowującej najmniejsze koszty przedostania się pomiędzy wierzchołkami i *predecessor[graph_size]*, przechowującej „poprzedników” wierzchołka, z którymi ma się najlepsze połączenie. Dzięki zastosowaniu tych dwóch tablic, łatwo można odtworzyć najlepszą ścieżkę i koszt jej przebycia.

Graf jest inicjowany krawędziami o losowej wadze z przedziału [1-20], łączącymi losowe wierzchołki z przedziału [0 - *graph_size()*]. Zagęszczenie grafu wynosi około 10%. Szukane połączenie to wierzchołek nr. 0 i wierzchołek nr *graph_size()/2*.

- **Tabele pomiarowe**

Niestety nie zdążyłem wykonać wszystkich pomiarów na czas. Sprawozdanie zostanie jak najszybciej uzupełnione.