Laboratorium 3

Łukasz Machnik 268456

# Zadanie 1 – opis algorytmów

Jako pierwszy zaimplementowałem algorytm Dijkstry. Jest to standardowy algorytm wyszukiwania najkrótszych ścieżek korzystający z kolejki priorytetowej opartej na standardowym kopcu. Algorytm zaczyna w wierzchołku źródłowym i wykonując BFSa sprawdza odległość od kolejnych wierzchołków. Złożoność tego algorytmu to gdzie n – liczba wierzchołków, m – liczba krawędzi.

Algorytm Diala zamiast zwykłej kolejki wykorzystuje kolejkę kubełkową. Dzieli ona kolejkę na kubełki, gdzie w każdym kubełku znajdują się tylko węzły z tym samym tymczasowym kosztem. Dzięki temu zabiegowi możliwa jest mniejsza złożoność czasowa wynosząca gdzie C to liczba kubełków która powinna być nie mniejsza niż maksymalna waga krawędzi.

Trzeci algorytm wykorzystuje RadixHeap – zmodyfikowany kopiec będący czymś pośrednim między tym z algorytmu Dijkstry a tym z algorytmu Diala. Wykorzystuje się w nim kubełki jednak pojedynczy kubełek może zawierać węzły o różnych kosztach będących jednak z pewnego ustalonego przedziału. Złożoność tego algorytmu to jest taka jak algorytmu Diala jednak mniejsza jest liczba kubełków – redukujemy ją do .

# Zadanie 2 – porównanie wyników

# Zadanie 3 - wnioski

Jak widać algorytm Dijkstry jest najbardziej czasochłonnym z nich wszystkich, jednak w zależności od konkretnego przypadku lepszy może być dial lub radix. Trzeba jednak pamiętać, że pomiary te są obarczone pewnym błędem, ponieważ czas wykonania może mocno zależeć od chwilowego obciążenia procesora.