

PAMSI

Projekt 3

Grafy

Magda Skoczeń

22.04.2022 - 13.05.2022

1 Zasady oddania projektu

- Należy oddać sprawozdanie opisujące wyniki eksperymentów (dotyczy zadania za 45 punktów (patrz sekcja 3) oraz za 60 punktów (patrz sekcja 4)) oraz kod źródłowy programu udostępniony na gitlabie + spakowany kod źródłowy wstawiony do eportalu.
- Program pisany w języku C++.
- Dla napisanej implementacji muszą przechodzić dostarczone testy.
- Termin oddania zadań: 13.05.2022 godzina 15.00. Każdy tydzień spóźnienia: -10 punktów.

1.1 Wykonywanie testów efektywności

Zadania poświęcone są badaniu efektywności algorytmów grafowych w zależności od metody reprezentacji grafów. Należy przygotować 2 reprezentacje grafów: w postaci macierzy sąsiedztwa oraz listy sąsiedztwa. Badania należy wykonać dla 5 różnych liczb wierzchołków w grafie V (np. 10, 50, 100, 500 i 1000) oraz następujących gęstości grafu: 25%, 50%, 75% oraz dla grafu pełnego. Dla każdego zestawu parametrów: algorytm, reprezentacja grafu, liczba wierzchołków i gęstość grafu należy wygenerować po 100 losowych instancji, natomiast w sprawozdaniu umieścić wyniki uśrednione.

1.2 Zawartość sprawozdania

dotyczy zadania za 45 punktów (patrz sekcja 3) oraz za 60 punktów (patrz sekcja 4). Sprawozdanie powinno zawierać:

- krótkie wprowadzenie
- opis badanych algorytmów z omówieniem ich złożoności obliczeniowej

- omówienie przebiegu eksperymentów i przedstawienie uzyskanych wyników (w postaci tabel i wykresów). Opis wymaganych wykresów w punkcie **1.2.1.**
- podsumowanie i wnioski (w przypadku niezgodności uzyskanych wyników z przewidywanymi spróbować wyjaśnić przyczyny)
- bibliografia (materiały wykorzystane do wykonania ćwiczenia, w tym strony internetowe)

1.2.1 Wykresy do sprawozdania

Wyniki należy przedstawić w tabelach oraz w formie wykresów. Wykresy powinny ilustrować zależność czasu wykonania algorytmu (oś Y) w funkcji ilości wierzchołków (oś X). Zaleca się przygotowanie następujących wykresów:

- Wykresy typu 1 (osobne wykresy dla każdej reprezentacji grafu) – wykresy, których parametrem jest gęstość grafu oraz typ algorytmu (czyli 4 gęstości grafu \times 2 typy algorytmów = 8 linii na rysunek).
- Wykresy typu 2 (osobne wykresy dla każdej gęstości grafu) – w formie linii których parametrem jest typ algorytmu i typ reprezentacji (czyli 4 linie na każdy rysunek).

2 Zadanie 30 punktów

Należy zaimplementować graf przechowujący elementy określonego typu. Należy napisać funkcje wykonujące podstawowe operacje na grafie zgodnie z informacjami podawanymi na wykładzie.

- Należy zaimplementować graf za pomocą listy sąsiedztwa zgodnie z wytycznymi na wykładzie
- Należy zaimplementować graf za pomocą macierzy sąsiedztwa zgodnie z wytycznymi na wykładzie.

Dodatkowo należy zaimplementować wczytywanie grafu z pliku tekstowego oraz możliwość wyświetlenia grafu na konsoli. Format pliku testowego to:

1. w pierwszej linii:

<i>liczba wierzchołków</i>	<i>liczba krawędzi</i>
----------------------------	------------------------
2. w kolejnych liniach:

<i>wierzchołek początkowy</i>	<i>wierzchołek końcowy</i>	<i>waga</i>
-------------------------------	----------------------------	-------------
3. w ostatniej linii:
wierzchołek startowy (dla problemu wyszukiwania najkrótszej ścieżki)

Wyświetlenie grafu należy zaimplementować w analogicznej formie.

Tylko dla osób oddających zadanie za 30 punktów!!

Należy zaimplementować możliwość wykonania operacji na grafie z poziomu konsoli tzn. użytkownik powinien móc wybrać reprezentację grafu, wczytać graf z pliku, wyświetlić wynikowy graf oraz wybrać aktualną operację na grafie z opisanych na wykładzie (np. dodanie krawędzi).

3 Zadanie 45 punktów - problem znajdowania minimalnego drzewa rozpinającego

1. Zaimplementować graf tak jak opisano w punkcie 2
2. Należy zaimplementować algorytmy Kruskala i Prima oraz przeprowadzić analizę efektywności tych algorytmów zgodnie z informacjami w punkcie 1.1.

Wynikiem działania algorytmu jest lista krawędzi wchodzących w skład znalezionej drzewa w formie:

wierzchołek 1 wierzchołek 2 waga

Wyniki dla przykładowych grafów znajdują się w folderze *mst_data*. Dokładną oczekiwaną implementację wyniku znajduje się w dołączonym kodzie źródłowym.

3. Przygotować program, który wczyta graf z podanego pliku do zadanej przez użytkownika implementacji grafu, wykona wskazany przez użytkownika algorytm i wyświetli wynik w formie jakiej został podany w przykładowych plikach wynikowych.
4. przygotować sprawozdanie według schematu z punktu 1.2.

4 Zadanie 60 punktów - problem znajdowania najkrótszej ścieżki w grafie

1. Zaimplementować graf tak jak opisano w punkcie 2
2. Należy zaimplementować algorytmy Dijkstry i Bellmana-Forda oraz przeprowadzić analizę efektywności tych algorytmów zgodnie z informacjami w punkcie 1.1. W przypadku algorytmu Dijkstry zaleca się implementację kolejki w formie kopca.

Wyniki algorytmów dla przykładowych grafów znajdują się w folderze *sp_data*. Wyniki zawierają najkrótsze ścieżki z zadanego źródła do wszystkich wierzchołków w postaci:

wierzchołek końcowy całkowita długość ścieżki lista kolejnych wierzchołków w ścieżce

3. Przygotować program, który wczyta graf z podanego pliku do zadanej przez użytkownika implementacji grafu, wykona wskazany przez użytkownika algorytm i wyświetli wynik w formacie jaki został podany w przykładowych plikach wynikowych.
4. przygotować sprawozdanie według schematu z punktu 1.2.

5 Źródła

- Jeleń Ł., Notatki do wykładu 7,8.
- Cormen T., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów, Wydawnictwo Naukowe PWN
- Drozdek A., C++. Algorytmy i struktury danych, Helion