Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Лабораторна робота №9

Виконав: Олексишин Олексій

Група К-28

Кафедра інтелектуальних програмних систем

**2020**

**Предметна область**

Вариант 3

Предметная область Отдел кадров Объекты Подразделения, Сотрудники Примечание Имеется множество подразделений предприятия. В каждом подразделении работает множество сотрудников.

Для того щоб дану структуру даних можна було помістити в дерево я перевантажив operator< (та інші оператори порівняння), для виведення в консоль operator<<.

**Завдання**

Реалізуйте алгоритм Джонсона для знаходження найкоротших шляхів в графі, у вершинах якого лежать об’єкти з предметної області.

**Алгоритм і теорія**

Алгоритм Джонсона використовує алгоритм Дейкстри та алгоритм Белмана-Форда для швидшого знаходження мінімільних відстаней між всіма парами вершин в зваженому орієнтованому графі. Працює навіть якщо в графі є ребра з від’ємною вагою, але немає циклів з від’ємною сумою ваг ребер. Ребра між вершинами зберігаються у вигляді списку суміжних вершин та відстаней між ними.

Алгоритм

Для початку, додамо до графа нову вершину V, та додамо ребра від неї до інших вершин. Далі запустимо процедуру алгоритма Белмана-Форда для даного графа та вершини V. Тепер за формулою weight[from][to] = weight[from][to] + bellman\_dists[from] – bellman\_dists[to] оновимо довжину всіх ребер у графі та видалимо вершину V і відповідні ребра. В кінці застосуємо для кожної вершини графа алгоритм Дейкстри.

**Складність алгоритму**

Складність алгоритму близька до складності алгоритму Флойда-Уоршелла – O(n^3), проте для розріджених графів (для яких кількість ребер m суттєво менша за квадрат кількості вершин) алгоритм матиме складність O(m\*n\*log(n)). Можна досягнути і меншої складності, але для цього замість звичайної черги з пріоритетом в алгоритмі Дейкстри треба використати фіббоначієву купу.

**Реалізація (мова С++)**

Програма реалізує клас Graph, який описує граф з вершинами що містять дані та списком ребер. Клас також містить функції реалізації алгоритмів Белмана-Форда, Дейкстри та власне алгоритма Джонсона

**Основні модулі програми**

У файлі graph.hpp показаний інтерфейс вказаних класів та реалізація методі для пошуку відстаней у графі (процедура dijkstra – реалізує алгоритм дейкстри, bellman\_ford – алгоритм Белмана-Форда, а процедура johnson\_pathes повертає матрицю з результатом роботи алгоритма Джонсона). Файл struct.hpp містить реалізацію класів предметної області. В файлі main.cpp міститься інтерфейс користувача та приклад роботи з реалізованою структурою даних.

Код можна знайти в моєму репозиторії на github за посиланням <https://github.com/ooleksyshyn/semester4/tree/master/algorithms/lab9>

**Інтерфейс користувача та приклади**

Програма дозволяє взаємодіяти з класом через простий інтерфейс – вставка, видалення, пошук елемента за ключем та. В коді є наведений приклад використання програми.

**Літературні джерела**

* <https://github.com/ooleksyshyn/semester4/tree/master/algorithms/lab9>
* <https://ru.wikipedia.org/wiki/Алгоритм_Дейкстры>
* <https://uk.wikipedia.org/wiki/Алгоритм_Джонсона>
* [https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%91%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0%E2%80%94%D0%A4%D0%BE%D1%80%D0%B4%D0%B0](https://uk.wikipedia.org/wiki/Алгоритм_Беллмана—Форда)