Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Алгоритми та складність

Завдання №ХХХ

“ Алгоритм Джонсона для розріджених графів ”

Варіант №ХХХ

Виконав студент 2-го курсу

Групи ХХХХХ

Сидоров Петро Іванович

Київ - 2023

**Завдання**:

Реалізація алгоритм Джонсона для розріджених графів (включає алгоритми Беллмана-Форда і Дейкстри). В алгоритмі Дейкстри використати піраміду Фібоначчі.

Предметна область: Аеропорт. Об'єкти: Авіакомпанії, Рейси

Примітка: Є безліч авіакомпаній. Для кожної авіакомпанії визначені її рейси.

**Теорія**

Алгоритм Джонсона дозволяє знайти найкоротші шляхи між усіма парами вершин зваженого орієнтованого графа. Цей алгоритм працює, якщо у графі містяться ребра з додатною чи від'ємною вагою, але відсутні цикли з від'ємною вагою.

Задача про найкоротший шлях полягає в знаходженні такого шляху між двома вершинами (або вузлами) графу, що сума ваг ребер з яких він складається мінімальна.

Зважений граф — граф, кожному ребру якого поставлено у відповідність деяке значення (вага ребра).

Розріджений граф – це граф, в якому кількість ребер наближається до кількості вершин: |Е | =О(| V|).

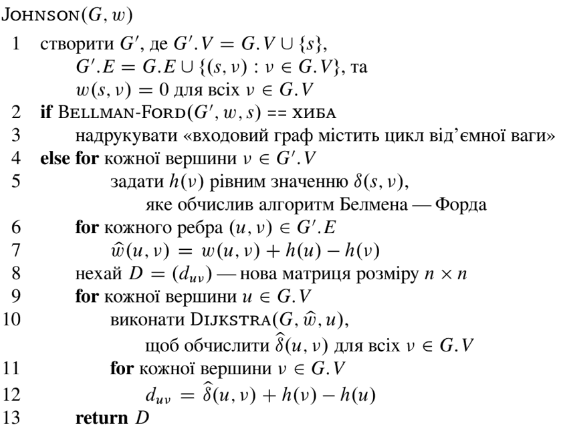
«Від'ємний цикл» - це такий цикл, сума ваги ребер якого дорівнює від'ємному значенню.

Алгоритм Беллмана-Форда — алгоритм пошуку найкоротшого шляху в зваженому графі. Знаходить найкоротші шляхи від однієї вершини графа до всіх інших. На відміну від алгоритму Дейкстри, алгоритм Беллмана—Форда допускає ребра з негативною вагою.

Алгоритм Дейкстри знаходить найкоротший шлях від однієї вершини графа до всіх інших вершин. Класичний алгоритм Дейкстри працює тільки для графів без циклів від'ємної довжини.

**Алгоритм**

Алгоритм Джонсона обчислює найкоротший шлях використовуючи алгоритм Беллмана-Форда та алгоритм Дейкстри, які він використовує у вигляді підпроцедур. Він неявно розрахований на те, що ребра зберігаються у списках суміжності. Алгоритм повертає вже звичайну матрицю D = *dij* розміру |V|\*|V|, де |V| - кількість вершин у графі або повідомляє про те, що вхідний граф містить цикл від’ємної ваги. Як типово для алгоритму пошуку найкоротших шляхів між усіма парами, вважаємо, що вершини пронумеровані від 1 до |V|. Цей код виконує описані вище дії:



**Складність алгоритму**

Якщо в алгоритмі Дейкстри неспадну чергу з пріоритетами втілено у вигляді піраміди Фібоначчі, то тривалість роботи алгоритму Джонсона дорівнює .

**Мова реалізації алгоритму** С++

**Модулі програми:**

* **class Airline**

Клас, що описує Авіакомпанії. Має параметр string name – назва компанії.

* **class Flight**

Клас, що описує Рейси. Має параметри string nameFlight – назва рейсу, Airline airline – авіакомпанія, int key - унікальний ключ для рейсу.

* **void Graph<T>::initSignleSource(size\_t main\_vertex)**

Метод скидає всіх батьків та ваги вершин, обнуляє вагу початкової вершини.

* **bool Graph<T>::relax(size\_t from, size\_t to, double weight)**

Метод ослаблює вершину. Повертає true, якщо ослаблення відбулося.

* **void Graph<T>::addVertex(T const &data)**

Метод додає нову вершину до графу. На вхід приймається рейс.

* **void Graph<T>::addEdge(size\_t from, size\_t to, double weight)**

Метод додає нове ребро до графу. На вхід індекс вершин на вага ребра.

* **bool Graph<T>::bellmanFordAlgorithm(size\_t init\_vertex)**

Алгоритм Беллмана-Форда для вказаної вершини.

**Складність:** .

* **void Graph<T>::dijkstraAlgorithm(size\_t init\_vertex)**

Алгоритм Дейкстри для вказаної вершини (на основі піраміди Фібоначчі).

**Складність:** .

* **std::pair<vector<vector<double>>,vector<vector<size\_t>>> Graph<T>::johnsonAlgorithm()**

Алгоритм Джонсона. Виводить матрицю мінімальних ваг вершин та матрицю оптимальних шляхів.

**Складність:** .

**Інтерфейс користувача**

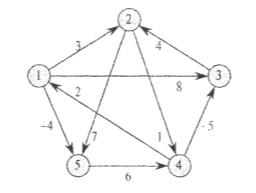
Введення даних відбувається через консоль. Спочатку на екран виводиться інформація про рейси, які було введено попередньо у коді.

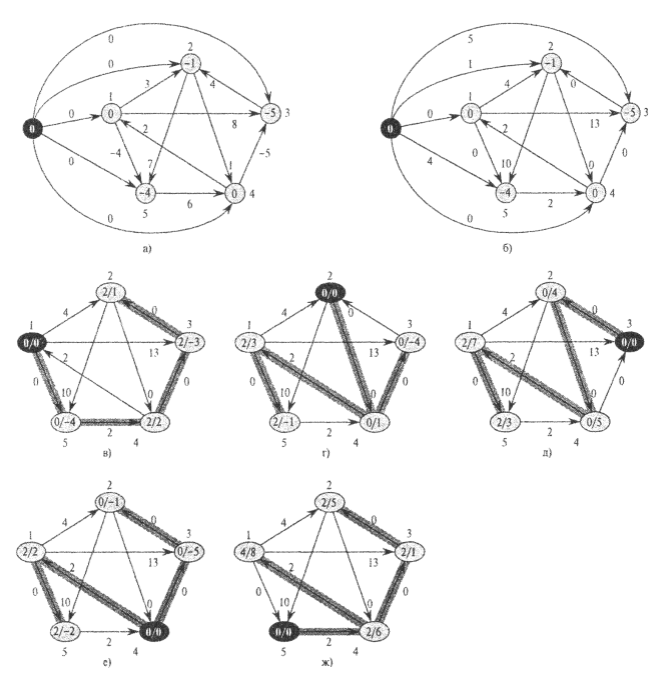
Вхідні дані: Номер операції яка вас цікавить. Доступні такі операції як

1. Додавання нової вершини за назвою рейсу
2. Додавання нового ребра
3. Виведення графу
4. Виведення матриці оптимальних ваг та шляхів за допомогою алгоритму Джонсона
5. Вихід з програми

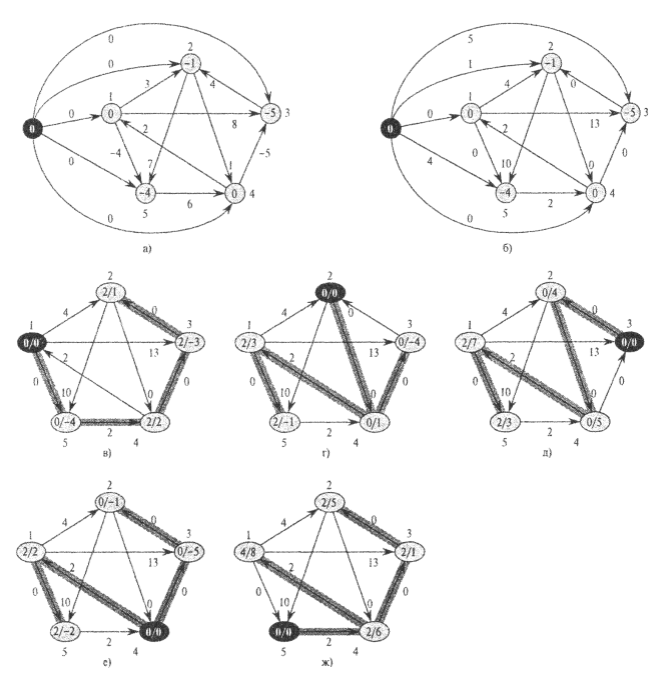
**Тестові приклади**

Перевіримо алгоритм Джонсона. Нехай маємо наступну множину рейсів та побудуємо граф зображений на рисунку знизу. Вершина вказує ідентифікаційний код кожного рейсу для зручності.

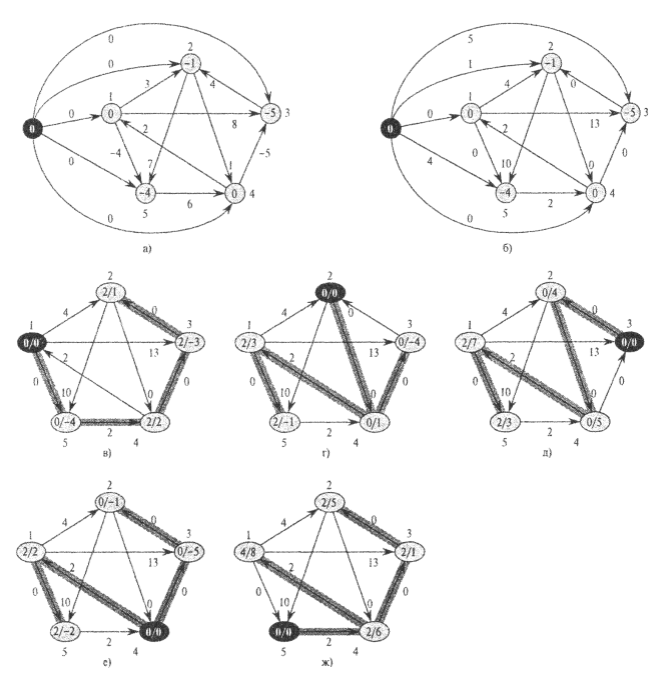




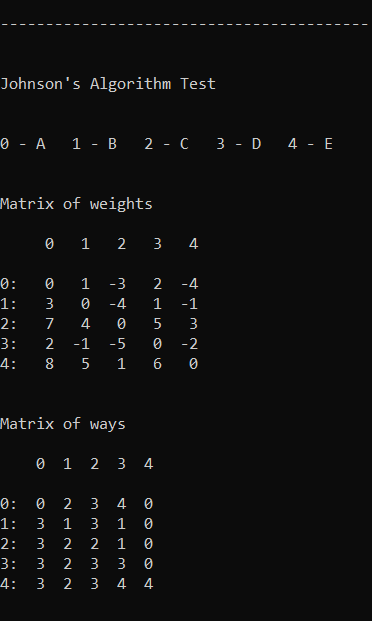
Спочатку створюється розширений граф з новою вершиною 0, що сполучена з усіма іншими вершинами, причому ребра мають нульову вагу;



потім з нової вершини запускається алгоритм Беллмана-Форда, який, окрім пошуку шляхів, перевіряє наявність циклів з від’ємною вагою, - таких циклів не виявлено;



далі значення ваг ребер запам’ятовуються та замінюються на інші – тепер від’ємних ваг немає; для кожної вершини запускається алгоритм Дейкстри, після нього значення мінімальних ваг вершин перераховуються та заносяться у відповідну матрицю, також заповнюється рядок матриці шляхів). Алгоритм повертає матриці ваг та шляхів.



Для орієнтування у таблиці ваг обирається рядок, що відповідає початковій вершині, та стовпчик, що відповідає кінцевій вершині. На їх перетині знаходиться шукана вага мінімального шляху.

Для орієнтування таблицею шляхів необхідно обирати рядок, що відповідає стартовій вершині, та стовпчик, що відповідає кінцевій вершині. На їх перетині вказаний стовпчик (вершина), куди варто перейти. Такі переходи здійснюються, доки стовпчик не почне переходити сам у себе, тобто коли досягнута кінцева вершина.

Результати роботи програми співпали з тестовим прикладом.

**Висновки**

Отже, у даній роботі, було описано алгоритм алгоритм Джонсона для розріджених графів (включає алгоритми Беллмана-Форда і Дейкстри). В алгоритмі Дейкстри використати піраміду Фібоначчі. Також було написано програму його реалізації, що повертає матрицю, яка містить ваги найкоротших шляхів для всіх пар вершин, або повідомляє, що вхідний граф містить цикл від’ємної ваги. У програмі доступні функції додавання нового елемента, пошуку та видалення елемента у дереві, а також виведення усіх версій червоно-чорного дерева.

Так як в алгоритмі Дейкстри використати піраміду Фібоначчі, то складність алгоритму Джонсона склала O(V2 log |V| + |V|\*|E|).

**Використані літературні джерела**

* «Алгоритми та складність». Лекція №10
* [https://en.wikipedia.org/wiki/Johnson%27s\_algorithm](https://en.wikipedia.org/wiki/Johnson%27s_algorithm%20)
* [https://en.wikipedia.org/wiki/Bellman–Ford\_algorithm](https://en.wikipedia.org/wiki/Bellman–Ford_algorithm%20)
* [https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm](https://en.wikipedia.org/wiki/Johnson%27s_algorithm%20)
* Т.Кормен, Ч.Лейзерсон, Р.Ривест, К.Штайн – «Алгоритмы. Построение и анализ». – 2013, [24 Розділ]