|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
|  | | |
| Домашнее задание № 2 | | |
| по дисциплине «Методы построения и анализа алгоритмов» | | |
|  | | |
| **Сортировка и поиск** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-81 |
| Варианты: | 7 |
| Студент: | Кондратьев Игорь |
|  |  |
|  |  |
| Преподаватель: | Георгий Анатольевич Щукин |
|  |  |
|  | | |
| Новосибирск | | |
| 2019 | | |

1. **Постановка задачи**  
   Реализовать класс “Граф” и его базовые операции, такие как вставка вершины или ребра/дуги, получение списка вершин смежных с заданной вершиной и т.д. Если в задании указан взвешенный и/или ориентированный граф, учесть это при проектировании (добавить возможность указания веса ребра и т.д.)  
     
   Дано взвешенный ориентированный граф. Необходимо найти кратчайший путь между всеми парами вершин с помощью алгоритма Флойда-Уоршелла
2. **Описание использованных алгоритмов**  
     
   **Алгоритм Флойда — Уоршелла** — [динамический](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5" \o "Динамическое программирование) алгоритм для нахождения кратчайших расстояний между всеми вершинами [взвешенного](https://ru.wikipedia.org/wiki/%D0%92%D0%B7%D0%B2%D0%B5%D1%88%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84" \o "Взвешенный граф) [ориентированного графа](https://ru.wikipedia.org/wiki/%D0%9E%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84" \o "Ориентированный граф)

Псевдокод:

for k = 1 to n

for i = 1 to n

for j = 1 to n

W[i][j] = min(W[i][j], W[i][k] + W[k][j])

1. **Текст реализации алгоритмов на языке программирования**

1. **Результаты тестирования**
2. **Результаты исследования времени работы алгортма**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во  узлов | 200000 | 300000 | 4000000 | 5000000 | 6000000 | 7000000 | 8000000 | 9000000 | 10000000 |
| Время (с) | 13.2 | 19.0 | 23.6 | 31.9 | 42.72 | 55.33 | 66.3 | 80.00 | 101.56 |

1. **Сравнение полученных результатов с ожидаемыми:**

Ожидаемые результаты для пирамидальной сортировки в среднем вычисляются, как **O((n/2)·log2n)**.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во  узлов | 200000 | 300000 | 4000000 | 5000000 | 6000000 | 7000000 | 8000000 | 9000000 | 10000000 |
| Время (с) | 13.2 | 19.0 | 23.6 | 31.9 | 42.72 | 55.33 | 66.3 | 80.00 | 101.56 |

Полученные результаты пирамидальной сортировки по фамилиям.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кол-во  узлов | 200000 | 300000 | 4000000 | 5000000 | 6000000 | 7000000 | 8000000 | 9000000 | 10000000 |
| Время (с) | 13.2 | 19.0 | 23.6 | 31.9 | 42.72 | 55.33 | 66.3 | 80.00 | 101.56 |

Полученные результаты близки с ожидаемыми. Это подтверждает представленные значения в таблицах и графиках, представленных выше.

1. **Вывод**  
   В процессе выполнения задачи я научился применять делегаты для сортировки по любому из признаку, провел модульное тестирование программы, а также построил графики, демонстрирующие зависимость времени работы от размера массива и сравение время работы написанной сортировки со встроенной сортировкой c# .NET. Полученные результаты совпадают с ожидаемыми/теоретическими.