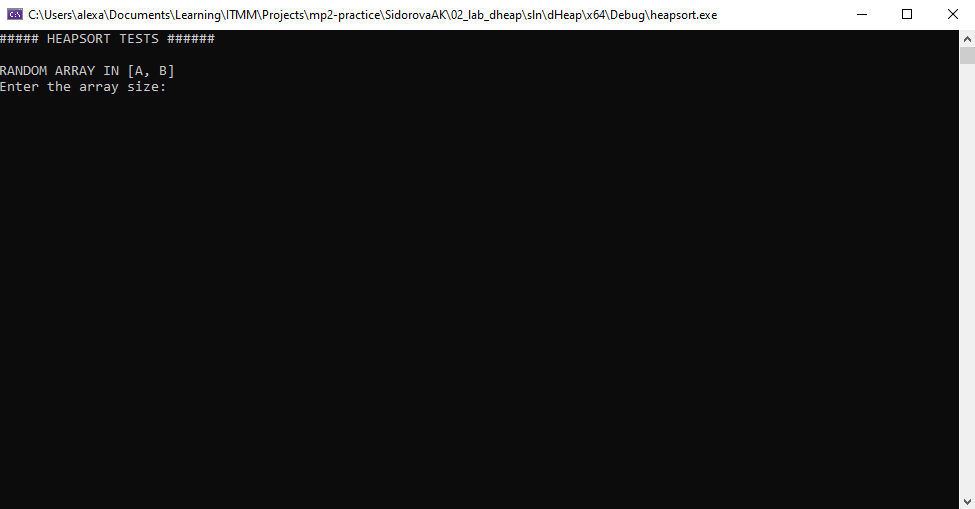
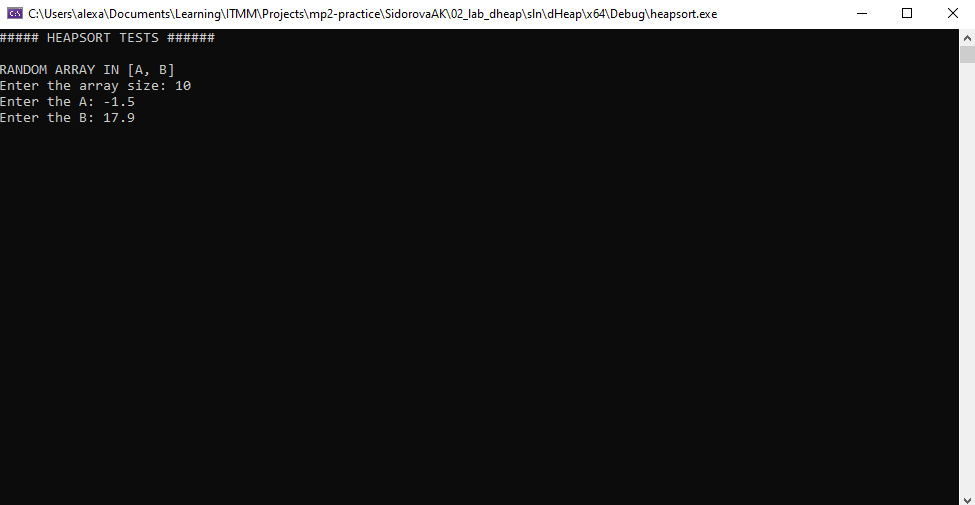
**Руководство пользователя**

# **1. HeapSort (Пирамидальная сортировка)**

В данном проекте реализовано тестирование работы алгоритма пирамидальной сортировки, который описан в статической библиотеке library.lib. При запуске программы откроется окно, в котором будут отображены результаты тестов.

Первый тест – сортировка массива, сгенерированного рандомно в заданном отрезке и с заданным размером. Пользователю будет предложено сначала ввести размер массива целым числом (Рис. 1).

**Рис. 1** Демонстрация работы программы.

Далее необходимо ввести соответственно левый и правый концы отрезка целым или вещественным числом (Рис. 2).

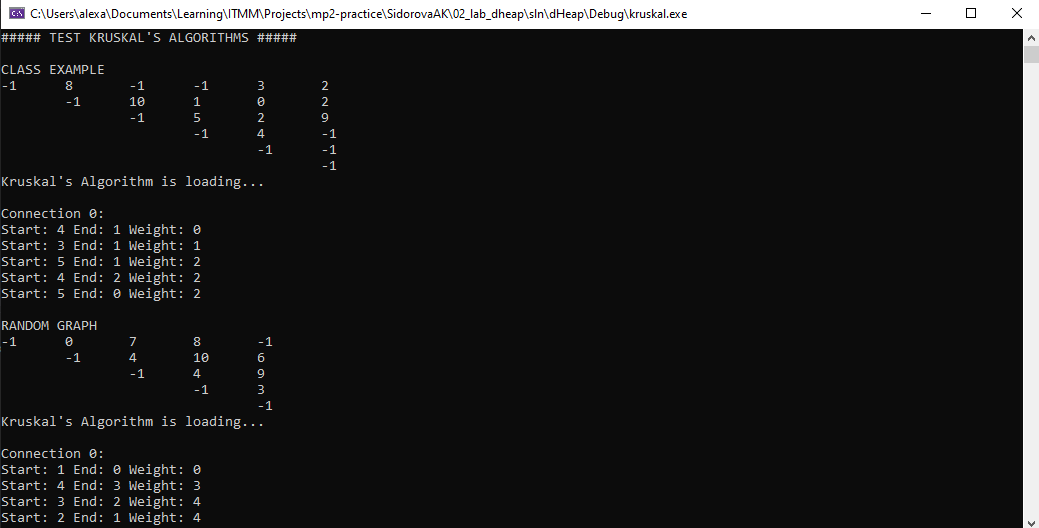
**Рис. 2** Демонстрация работы программы.

Таким образом, сначала отобразится сгенерированный массив, а после сразу же отсортированный в порядке убывания (Рис. 3).

**Рис. 3** Демонстрация работы программы.

Кроме того, сразу же будут выведены результаты остальных двух тестов: попытка отсортировать пустой массив и попытка отсортировать массив с указанным отрицательным размером. Но будут выведены ошибки, поскольку отсортировать нечего, а размер не может быть отрицательным.

# **2. Kruskal (Алгоритм Краскала)**

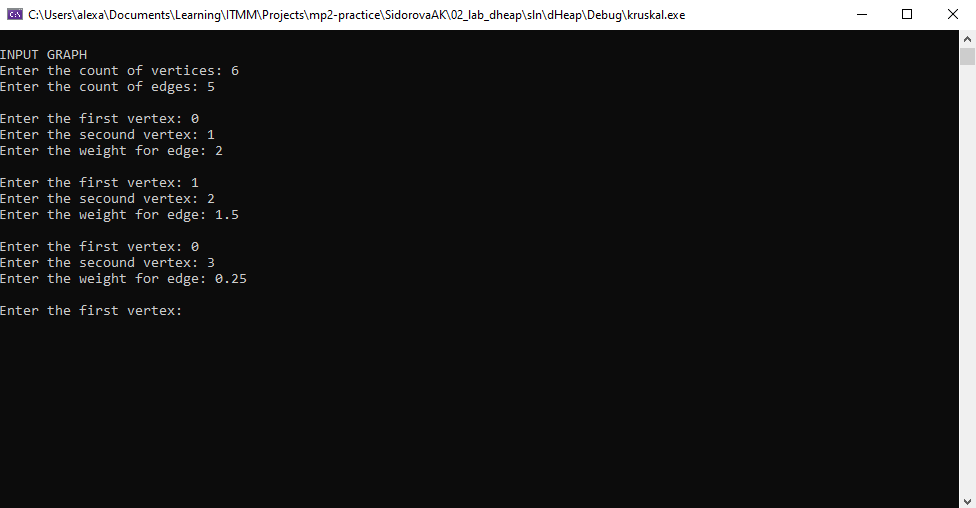
Данный проект содержит в себе файл с тестами работы алгоритма Краскала. При запуске программы пользователь увидит тест с графом, который был разобран в аудитории на паре, чтобы сравнить результаты работы программы и результаты, полученные в аудитории (Рис. 4).

**Рис. 4** Демонстрация работы программы

Сначала выводится верхне-треугольная матрица смежности графа, ведь ее достаточно, чтобы определить смежные ли вершины и какой вес у определенного ребра. Стоит отметить, что «-1» - определяет отсутствие ребра, а любое другое неотрицательное число – вес ребра между данными вершинами (номер столбца и строчки).

Затем для каждого класса смежности («Connection #») выводится список ребер в формате: стартовая вершина, конечная вершина, вес между данными вершинами. Таким образом, для примера из аудитории получается 1 класс смежности с 5 ребрами наименьшего размера, не образующие цикл.

Далее на том же рисунке можно увидеть второй тест для графа, который сгенерирован рандомно, и результаты теста по этому графу.

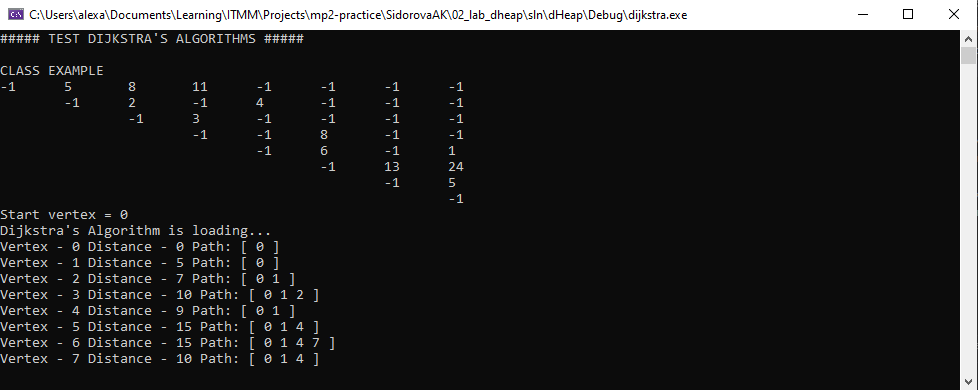
Кроме того, в конце пользователю будет предложено ввести свой граф. Сначала необходимо ввести положительное целое число вершин, затем положительное целое число ребер. После чего будет предложено вести информацию о ребрах в формате: стартовая вершина, конечная вершина, вес. Стоит отметить, что нумерация вершин идет с нуля, а вес должен быть неотрицательным (Рис. 5).

**Рис. 5** Демонстрация работы программы.

В итоге, по окончании заполнения информации о графе будет выведен граф в виде верхне-треугольной матрицы смежности, а затем результаты тестирования (Рис. 6). В нашем случае получилось 2 класса смежности и, следовательно, по каждому классу получили свое дерево (каркас).

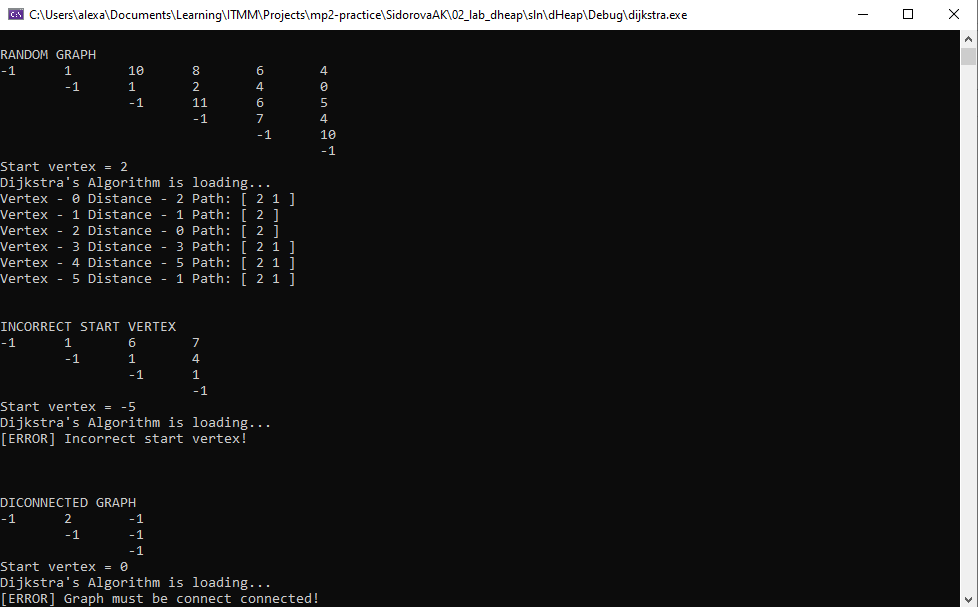
**Рис. 6** Демонстрация работы программы.

# **3. Dijkstra (Алгоритм Дейкстры)**

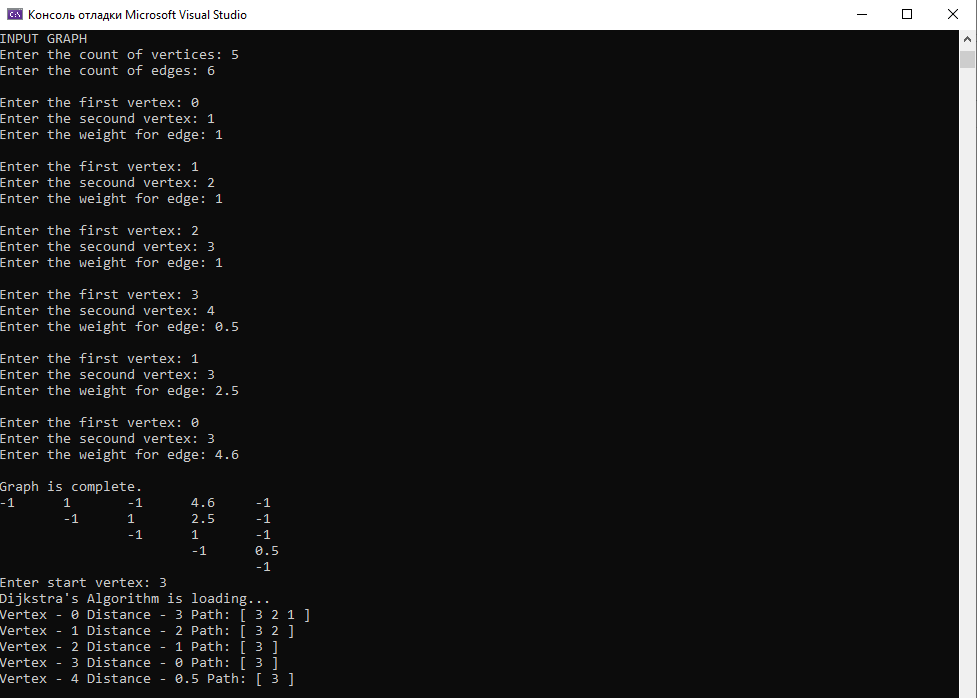
В данном проекте реализованы тесты работы алгоритма Дейкстры. При запуске программы первым делом можно увидеть первый тест графа, который был разобран в аудитории на паре, и, следовательно, сравнить результаты работы программы и результаты, полученные в аудитории (Рис. 7).

**Рис. 7** Демонстрация работы программы.

Сначала будет выведена верхне-треугольная матрица смежности, затем номер стартовой вершины (нумерация идет с нуля!), а после список вершин с результатами по каждой в формате: номер вершины, кратчайшее расстояние от стартовой до данной вершины, путь от стартовой до данной, не включая ее саму (за исключением стартовой). Таким образом, можно восстановить все пути от стартовой до определенной вершины.

Затем идет тестирование рандомно сгенерированного связного графа и с рандомно сгенерированной стартовой вершиной, тестирование с отрицательной стартовой вершиной, тестирование алгоритма Дейкстры на несвязанной графе (Рис. 8).

**Рис. 8** Демонстрация работы программы.

И в самом конце будет предложено самому пользователю ввести свой связанный граф в таком же формате, как и в алгоритме Краскала. После отображения верхне-треугольной матрицы смежности необходимо ввести номер стартовой вершины. Затем идет работа алгоритма Дейкстры и соответственно вывод результатов его работы (Рис. 9).

**Рис. 9** Демонстрация работы программы.