

# Руководство пользователя

## 1. Пирамидальная сортировка

При запуске программы необходимо ввести количество сортируемых элементов. Если введенное число будет меньше 0, программа попросит вас ввести число заново.

```
-Heapsort-  
Enter size of array -10  
Enter size of array _
```

Если размер массива введен верно, далее необходимо будет ввести диапазон генерируемых чисел. Далее первой строкой будут показаны генерируемые числа, второй строкой отсортированный в порядке убывания массив чисел.

```
-Heapsort-  
Enter size of array -10  
Enter size of array 10  
Enter beginning of range -5  
Enter ending of range 50  
40 7 40 17 15 17 10 13 39 38  
40 40 39 38 17 17 15 13 10 7
```

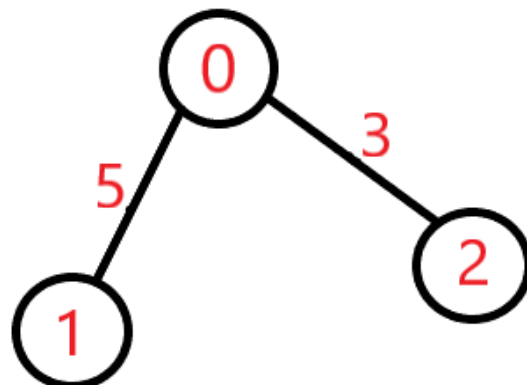
## 2. Алгоритм Краскала

При запуске программы необходимо создать желаемый граф. Для этого введите количество вершин и ребер.

```
-Kruskal's Algorithm-  
Enter count of vertexes: 3  
Enter count of edges: 2
```

Далее пошагово создаем ребра. Сначала вводим начало ребра, потом конец ребра и вес ребра. Если какие-то данные будут введены некорректно, программа заново предложит вам создать текущее ребро.

```
-Kruskal's Algorithm-  
Enter count of vertexes: 3  
Enter count of edges: 2  
  
Edge 0  
Begin: 0  
End: 1  
Weight = 5  
  
Edge 1  
Begin: 0  
End: 10  
Weight = 3  
  
Edge 1  
Begin: 0  
End: 2  
Weight = 3
```



Если граф не связный, программа выведет сообщение о данной ошибке. При попытке создать петлю в графе, программа предложит создать ребро заново.

```
-Kruskal's Algorithm-
Enter count of vertexes: 3
Enter count of edges: 1

Edge 0
Begin: 0
End: 1
Weight = 5

Graph isn't connected!
```

После создания графа будет выведен результат алгоритма Краскала – оставное дерево. Оно будет выведено в столбик с помощью ребер.

```
Ostav Tree:
0 - 2 W = 3
0 - 1 W = 5
```

### 3. Алгоритм Дейкстры

Как и в предыдущем алгоритме, перед началом работы необходимо создать граф.

```
Enter count of vertexes: 4
Enter count of edges: 5

Edge 0
Begin: 0
End: 1
Weight = 9

Edge 1
Begin: 0
End: 2
Weight = 1

Edge 2
Begin: 2
End: 3
Weight = 2

Edge 3
Begin: 1
End: 3
Weight = 3

Edge 4
Begin: 0
End: 3
Weight = 10
```

Далее необходимо выбрать начальную вершину, от которой будет выполняться поиск кратчайших расстояний. При некорректном вводе вершины, программа предложит заново ввести верное число. Далее будет выведен результат работы программы.

```
Enter starting vertex: -1
Enter starting vertex: 0

Vertex 0 path  0      W = 0
Vertex 1 path  0 2 3   W = 6
Vertex 2 path  0      W = 1
Vertex 3 path  0 2     W = 3
```