Section: CPS3410

Group Member: Hu Hanxing, Fu Dongfang

IDs: not known, 987757

Title: Project Document

**Instruction**

This is a project implementing priority queue in two different algorithm. Priority queue is to arrange a queue according to the priority of each element in queue. The older version of priority queue is to sort the elements in an array according to the priority of each element. The new version is to store the elements into a minimum heap. Elements with lower priority will be put at the front positions of the queue.

**Pseudo-code**

1. **Old Version:**
2. **New Version:**

**Insert{**

**Check whether array need to be expended;**

**Insert value at bottom of array;**

**Adjust Heap;**

**Size++;**

**}**

**Delete\_Min{**

**Check Empty;**

**Swap queue[1] and queue[size];**

**Queue[size]=0;**

**Size--;**

**Adjust Heap;**

**}**

**FindMin{**

**Check Empty;**

**Return queue[1];**

**}**

**isEmpty{**

**return size <= 0;**

**}**

**Evaluation**

**Big-O Summary Table**

|  |  |  |
| --- | --- | --- |
| Function | Time Complexity (Old) | Time Complexity (New) |
| insert | O(N) |  |
| delete\_min | O(N) |  |
| isEmpty | O(1) |  |
| findMin | O(1) |  |
| deleteMin | O(N) |  |

**Old Version:**

**New Version:**

Every time when inserting, it will put the new value at the bottom of the array. Then it will adjust the heap to build minimum heap. Each time adjusting the heap, the time complexity will be . Each time when deleting the minimum value, it will also adjust the heap.