CS222 - Algorithm Design

Dr. Arpita Korwar

Assignment – 3 : Heapsort

Authors :

- Prakhar Mathur 1906328
- Sanjay Marreddi 1904119
- Rishabh Tripathi 1904129

Problem Statement

1. :

   (a) Implement the heap data structure as a C++ class. The public methods should include:

```
class maxHeap{
private:
        int* items;
        int size;

        int sift_down(int); //takes an index. Assume all the other
            ↪ items in the sub-tree rooted at the index satisfy the
            ↪ max-heap property. Swap items in the sub-tree rooted
            ↪ at the index, so that the sub-tree rooted at the index
            ↪  is a max-heap.
        int sift_up(int); //takes an index. Assume all the other
            ↪ items in the heap satisfy the max-heap property. Swap
            ↪ items from the root to the index so that the tree
            ↪ satisfies the max-heap property.
        int find_parent(int); //Returns the index of the parent
        int find_lchild(int);//Returns the index of the left child
        int find_rchild(int);//Returns the index of the right child
        bool is_valid_index(int);
public:
```

```
        void heapify(); //called by the constructor below to convert
            ↪ the arbitrary array items into a max-heap.
        int get_max (); //peeks into the max-heap and returns the
            ↪ maximum value
        void insert(); //inserts an element into the max heap
        int delete_max(); // returns the maximum value and deletes
            ↪ the item.

            maxHeap(int sz, int arr[]){
            //Constructor that takes an arbitrary array of size
                ↪ sz and creates a max-heap.
            }
};
```

You may add other private and public methods.

(b) Write a function `heapsort` that sorts an input array using an object of `maxHeap`.

(c) In the main method, create a random array of length 20 and call heapsort on that.

(d) in a pdf file, write:

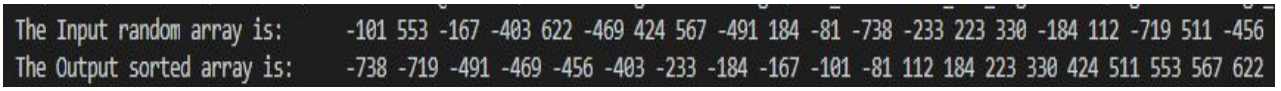
1. the time complexity of each of the methods in the class `maxHeap`,
2. the time complexity of heapsort,
3. the random array and the output.

c) In the main function to randomize array **rand()** function has been used.

d)

1. Time complexity of
    - sift_down : O(log(n))
    - sift_up : O(log(n))
    - find_parent : O(1)
    - find_lchild : O(1)
    - find_rchild : O(1)
    - is_valid_index : O(1)
    - heapify : O(log(n))
    - get_max : O(1)
    - insert : O(log(n))
    - delete_max : O(log(n))

2. Time complexity of heap sort is **O(nlog(n))**.
3. The Input Array was randomly generated. One instance of Input and Output are as follows:



--- The End ---