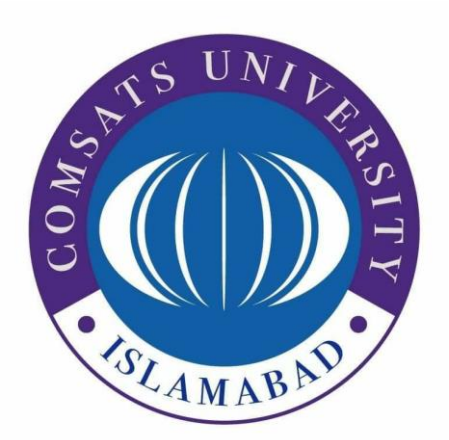

DATA STRUCTURES AND ALGORITHM

Lab final report

Name: Tayyaba

Registration Number: FA21-BCE-093

Instructor: Omar Ahmad

Binary Heap Implementation:

1. **Complete the function 'void insert_node (int x, struct heap_struct * H)' in the skeleton code provided.**

   In this task, firstly check whether the array is completely filled or not and then increase the size of Heap array in order to insert new data. Algorithm is written for the implementation of max heap. So, WHILE loop is used to check whether the inserted node have value greater or lesser than parent node. If the inserted node is greater than parent node, than swap the values and again check the condition until it goes to top most root node.

2. **Complete the function 'void delete_root (struct heap_struct * H)' in the skeleton code provided**.

   In this task, we firstly check if the array is empty. The root node is eliminated first. The final piece of the previous level is then transferred to the root. This child node's value is contrasted with that of its parent. The elements are switched if the value of the parent is smaller than that of the child.
   If the parent is less than the left child, the while loop is utilized with the condition checking. If the answer is yes, the child node is swapped. Then, a different criterion is applied to see if the parent is not the child's ideal parent. If so, switch them out such that the node is equivalent to the child node. The child index must also fit inside the current heap size in order to avoid containing trash values.

Critical analysis:

In this lab we learn the implementation of insertion and deletion operation in max heap. We came to know about the limitations and benefits of using array for insertion of data in heaps. The array is limited by its predefined size while it is very convenient to find the parent node in predefined array.