LAB # 12

Follow the steps to build a Heap data structure in C++

Objective

- To understand the Heap Data Structure
- To implement maxheap and minheap

Pre-Lab

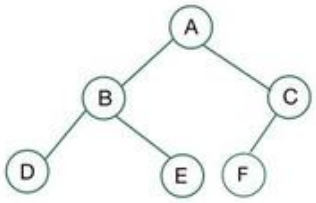
What is Heap Data Structure?

A heap is a specialized tree-based data structure that satisfies the heap property: If A is a parent node of B then the key of node A is ordered with respect to the key of node B with the same ordering applying across the heap. Either the keys of parent nodes are always greater than or equal to those of the children and the highest key is in the root node (this kind of heap is called max heap) or the keys of parent nodes are less than or equal to those of the children and the lowest key is in the root node (min heap).

Heaps are crucial in several efficient graph algorithms such as Dijkstra's algorithm, and in the sorting algorithm heapsort. A min-max heap is a complete binary tree containing alternating min and max levels

Example of Min-max heap : If it is not empty, each element has a data member called key. The root is always present at min level. Let x be any node in a min-max heap. If x is on a min (max) level then the element in x has the minimum (maximum) key from among all elements in the subtree with root x. A node on a min (max) level is called a min (max) node.

A complete binary tree is a special type of binary tree where all the levels of the tree are filled completely except the lowest level nodes which are filled from as left as possible.



Complete Binary Tree

Types of heaps:

Generally, heaps are of two types.

1) **Max-Heap:**

   In this heap, the value of the root node must be the greatest among all its child nodes and the same thing must be done for its left and right sub-tree also.

2) **Min-Heap:**

In this heap, the value of the root node must be the smallest among all its child nodes and the same thing must be done for its left and right sub-tree also.

Operations Supported by Heap:

Operations supported by min – heap and max – heap are same. The difference is just that min-heap contains minimum element at root of the tree and max – heap contains maximum element at the root of the tree.

a) Heapify:

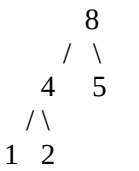
It is the process to rearrange the elements to maintain the property of heap data structure. It is done when a certain node creates an imbalance in the heap due to some operations on that node. It takes O(log N) to balance the tree.

b) Insertion:

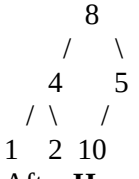
If we insert a new element into the heap since we are adding a new element into the heap so it will distort the properties of the heap so we need to perform the heapify operation so that it maintains the property of the heap.

Examples:

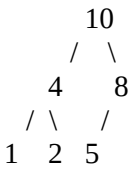
Assume initially heap(taking max-heap) is as follows



Now if we insert 10 into the heap



After **Heapify** operation final heap will be look like this



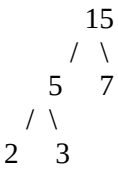
c) Deletion:

If we delete the element from the heap it always deletes the root element of the tree and replaces it with the last element of the tree.

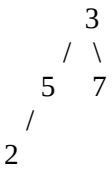
Since we delete the root element from the heap it will distort the properties of the heap so we need to perform Heapify operations so that it maintains the property of the heap.

Example:

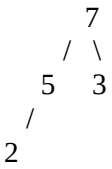
Assume initially heap(taking max-heap) is as follows



Now if we delete 15 into the heap it will be replaced by leaf node of the tree for temporary.



After Heapify operation final heap will be look like this



Pre-Lab Task

Task 1: Create an algorithm for construction of a minheap using Arrays

In-Lab Tasks

Lab Task 1: Write a C++ program that implements minheap in C++

Write a C++ code, that takes 10 inputs from the user and implement a minheap using Arrays data structures

Lab Task2: Write a C++ program that modify a maxheap in C++

Write a C++ code that implements delete operation in a maxheap. Maxheap is implemented using Arrays

- For both tasks, you should use functions wherever needed
- Your code must be menu driven (code only exits when -1 is entered as input)

Rubric for Lab Assessment

The student performance for the assigned task during the lab session was:			
Excellent	The student completed assigned tasks without any help from the instructor and showed the results appropriately.	4	
Good	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.	3	
Average	The student could not complete all assigned tasks and showed partial results.	2	
Worst	The student did not complete assigned tasks.	1	

Instructor Signature: _____ Date:_____