

ASSINGMENT NO.	10
TITLE	Implement the Heap sort algorithm
PROBLEM STATEMENT /DEFINITION	Implement the Heap sort algorithm using heap data structure with modularity of programming language.
OBJECTIVE	1. To understand and implement concept of heap in data structure.
OUTCOME	<ul style="list-style-type: none"> • Implement heap using Object Oriented features. • Implement heap sort and analyze its working
S/W PACKAGES AND HARDWARE APPARATUS USED	<ul style="list-style-type: none"> • 64-bit Open source Linux or its derivative. • Open Source C++ Programming tool like G++/GCC.
REFERENCES	Data structures in C++ by Horowitz, Sahni.
INSTRUCTIONS FOR WRITING JOURNAL	<ol style="list-style-type: none"> 1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 9. Conclusion/Analysis

Prerequisites: Understanding and implementation of complete binary tree

Learning Objectives:

To understand and implement heap and heap sort in data structure.

Learning Outcomes

After successful completion of this assignment, students will be able to

- Implement heap using Object Oriented features.
- Implement heap sort and analyze its working

Concepts related Theory:

Let us first define a Complete Binary Tree. A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible

A Binary Heap is a Complete Binary Tree where items are stored in a special order such that value in a parent node is greater (or smaller) than the values in its two children nodes. The former is called as max heap and the latter is called min heap. The heap can be represented by binary tree or array.

Why array based representation for Binary Heap?

Since a Binary Heap is a Complete Binary Tree, it can be easily represented as array and array based representation is space efficient. If the parent node is stored at index I , the left child can be calculated by $2 * I + 1$ and right child by $2 * I + 2$ (assuming the indexing starts at 0).

Basic Operations:

Heap Sort Algorithm for sorting in increasing order:

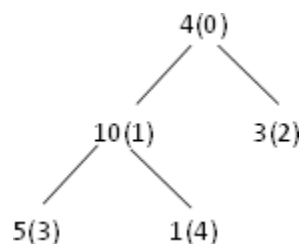
1. Build a max heap from the input data.
2. At this point, the largest item is stored at the root of the heap. Replace it with the last item of the heap followed by reducing the size of heap by 1. Finally, heapify the root of tree.
3. Repeat above steps until size of heap is greater than 1.

How to build the heap?

Heapify procedure can be applied to a node only if its children nodes are heapified. So the heapification must be performed in the bottom up order.

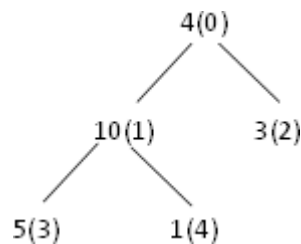
Lets understand with the help of an example:

Input data: 4, 10, 3, 5, 1



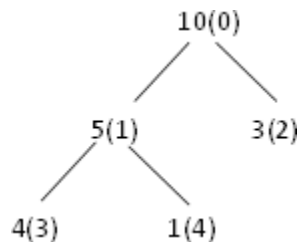
The numbers in bracket represent the indices in the array representation of data.

Applying heapify procedure to index 1:



Applying heapify procedure to index 0:

The heapify procedure calls itself recursively to build heap in top down manner.



Algorithm:

Input: Number of elements to be sort and element values.

Output: Elements in sorted order.

STEP 1: Logically, think the given array as Complete Binary Tree,

STEP 2: For sorting the array in ascending order, check whether the tree is satisfying Max-heap property at each node, (For descending order, Check whether the tree is satisfying Min- heap property) Here we will be sorting in Ascending order,

STEP 3: If the tree is satisfying Max-heap property, then largest item is stored at the root of the heap. (At this point we have found the largest element in array,

Now if we place this element at the end(nth position) of the array then 1 item in array is at proper place.)

We will remove the largest element from the heap and put at its proper place(nth position) in array.

After removing the largest element, which element will take its place? We will put last element of the heap at the vacant place. After placing the last element at the root, The new tree formed may or may not satisfy max-heap property. So, If it is not satisfying max-heap property then first task is to make changes to the tree, So that it satisfies max-heap property.

(Heapify process: The process of making changes to tree so that it satisfies max-heap property is called heapify)

When tree satisfies max-heap property, again largest item is stored at the root of the heap. We will remove the largest element from the heap and put at its proper place(n-1 position) in array. Repeat step 3 until size of array is 1 (At this point all elements are sorted.)

Conclusion: This program gives us the knowledge of heap data structure

Review Questions:

- What is a heap in data structure?
- What is complexity of heap sort?
- What is min heap and min heap?
- Explain heapify operation?