

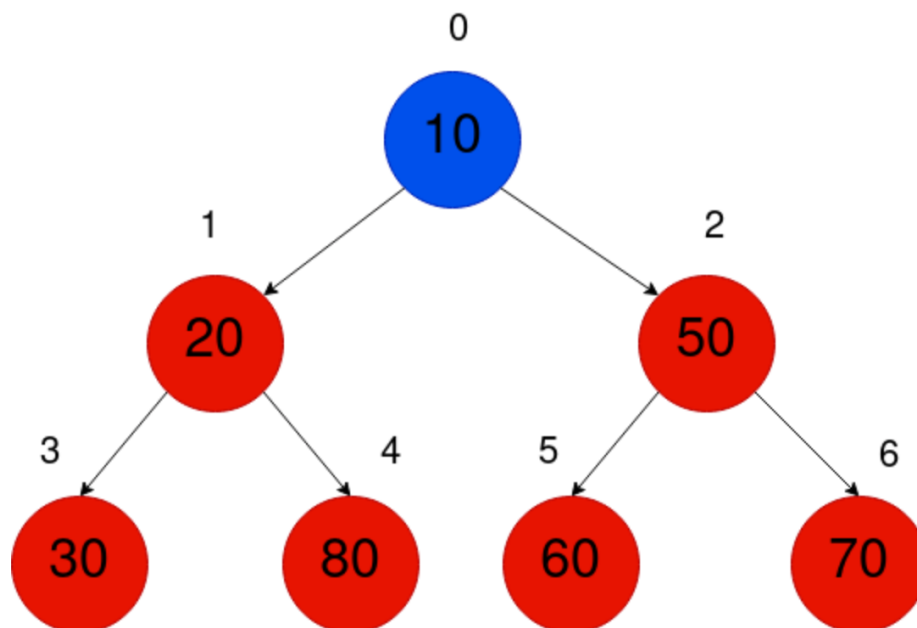
- Assignment Name: Lab6
- Description: Question based on Tree (Min Heap)
- Total Marks: 10
- Deadline: Depends on respective lab hrs
- Problem Title: **Implement Bottom up heap construction**
- Marks allotted: 10
- Description:

Create a min heap out of an array of N integers of a certain size S. You will be implementing a bottom up approach to this.

A Min Heap is a Tree where the root node has the most minimum key in the tree. Every node is smaller than its children.

The above definition holds true for all sub-trees in the tree. This is called the Min Heap property. It is important to note that heaps are **not always sorted**, the key condition that they follow is that the largest or smallest element is placed on the root node (top) depending if it is a Max or Min Heap.

The below tree is an example of a min heap tree.



The numbers marked above are the indexes of an array where the element would be placed and read in level order traversal. The array maintains a 0-based index. For example, this would be the array representation of the above min heap.

0	1	2	3	4	5	6
10	20	50	30	80	60	70

You are required to implement the following tree operations as a part of this lab exercise:

- 1) void create_minheap(MinHeap *heap)
 - A function that takes in a struct heap, which includes an array and size of that array, and adjusts the heap in a bottom up manner from the last parent index (size/2) up to the root, by comparing it with children.
- 2) void print_minheap(MinHeap* heap);
 - Implement this function that traverses all elements in the final constructed heap in the form of an array, **until the size of the array**. You needn't call this function anywhere, it is handled by the main program. You only need to implement it. **Do not call it anywhere else.**

You are supposed to use the code provided to you via the boilerplate code.

The skeleton of the min heap is provided to you via “struct” so you need not code that.

```
typedef struct MinHeap MinHeap;
struct MinHeap {
    int* arr;
    int size; //size of the heap
};
```

Functions to initialise minheap, and free min heap has already been implemented in the boilerplate code.

USING THE BOILERPLATE CODE PROVIDED IS MANDATORY.

- Input format:
 - Each test case starts with an input of 2 lines.
 - First line of input consists of an integer N, representing the size of the array, which is the number of elements that it holds.
 - Second line consists of an array of space separated integers out of which you'll be constructing a min heap.
- Output format:
 - The output will be an array of space separated integers which is actually the level order traversal of the min heap printed using the **print_heap** method.
- Sample test cases:
 1. TC #1:
 - Input:
4
3 2 4 1
 - Expected output:
1 2 4 3
 2. TC #2:
 - Input:
4
5 3 1 2
 - Output:
1 2 5 3