**Heap Sort :-**

Medium Accuracy: 53.06% Submissions: 84K+ Points: 4

Given an array of size N. The task is to sort the array elements by completing functions **heapify**() and **buildHeap**() which are used to implement Heap Sort.

**Example 1:**

**Input:**

N = 5

arr[] = {4,1,3,9,7}

**Output:**

1 3 4 7 9

**Explanation:**

After sorting elements

using heap sort, elements will be

in order as 1,3,4,7,9.

**Example 2:**

**Input:**

N = 10

arr[] = {10,9,8,7,6,5,4,3,2,1}

**Output:**

1 2 3 4 5 6 7 8 9 10

**Explanation:**

After sorting elements

using heap sort, elements will be

in order as 1, 2,3,4,5,6,7,8,9,10.

**Your Task** **:**  
You don't have to read input or print anything. Your task is to complete the functions **heapify()***,***buildheap()** and **heapSort()**where heapSort() and buildheap() takes the array and it's size as input and heapify() takes the array, it's size and an index i as input. Complete and use these functions to sort the array using heap sort algorithm.  
**Note:**You don't have to return the sorted list. You need to sort the array "arr" in place.

**Expected Time Complexity:** O(N \* Log(N)).  
**Expected Auxiliary Space:** O(1).

**Constraints:**  
1 ≤ N ≤ 106  
1 ≤ arr[i] ≤ 106

**Code :-**

//{ Driver Code Starts

// C++ program for implementation of Heap Sort

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// The functions should be written in a way that array become sorted

// in increasing order when heapSort() is called.

class Solution

{

public:

//Heapify function to maintain heap property.

void heapify(int arr[], int n, int i){

int left=2\*i + 1, right=2\*i + 2, large=i;

if(left<n && arr[large]<arr[left]) large=left;

if(right<n && arr[large]<arr[right]) large=right;

if(arr[large] != arr[i]){

swap(arr[large], arr[i]);

heapify(arr, n, large);

}

return;

}

public:

//Function to build a Heap from array.

void buildHeap(int arr[], int n){

for(auto i=(n/2)-1; i>=0; i--){

heapify(arr, n, i);

}

return;

}

public:

//Function to sort an array using Heap Sort.

void heapSort(int arr[], int n){

buildHeap(arr, n);

for(auto last=n-1; last>=0; last--){

swap(arr[0], arr[last]);

heapify(arr, last, 0);

}

return;

}

};

//{ Driver Code Starts.

/\* Function to print an array \*/

void printArray(int arr[], int size)

{

int i;

for (i=0; i < size; i++)

printf("%d ", arr[i]);

printf("\n");

}

// Driver program to test above functions

int main()

{

int arr[1000000],n,T,i;

scanf("%d",&T);

while(T--){

scanf("%d",&n);

for(i=0;i<n;i++)

scanf("%d",&arr[i]);

Solution ob;

ob.heapSort(arr, n);

printArray(arr, n);

}

return 0;

}

// } Driver Code Ends

**T.C :- O(n \* log n)**

**S.C :- O(1) without recursive call stack**