
Delete a node from a MAX-HEAP.

By Ajay Pilania (IRM2014007)
Shubham Swarnkar (IWM2014001)
Ashok Maloth (IIM2014006)

Introduction

For a given MAX-HEAP delete any arbitrary node from a Max-Heap (not only top most node).

Method 1 -:

Extract_node (Heap , index , size) :

 ret = Heap[index]

 Heap[index] = INT_MAX

 parent = index / 2

 While (parent > 0) do :

 swap(Heap[index] , Heap[parent])

 index = parent

 parent = index/2

Delete_top(Heap , size)

```
Delete_top (Heap , size):  
    IF size == 0  
        Return  
    Ret = Heap[1]  
    Swap(Heap[1] , Heap[size])  
    size = size - 1  
    Heapify (Heap , size , 1)  
    Return Ret
```

```
Heapify (Heap , size , index):  
    IF index > size  
        Return  
    S = index;  
    L = index*2;  
    R = L+1;  
  
    IF L <= size AND heap[L] > heap[S]:  
        S = L;  
    IF R <= size AND heap[R] > heap[S]:  
        S = R;  
    IF S not equal index:  
        Swap(heap[index] , heap[S])  
        Heapify(Heap , size , S)
```

Method 2 -:

```
Extract_node (Heap , index , size) :  
    ret = Heap[index]  
    Heap[index] = Heap[size]  
    size = size-1  
    RestoreUP(Heap , size , index)  
    Heapify(Heap,size,index) //same as above  
    Return ret
```

```
RestoreUP(Heap , size , index) :  
    parent = index/2;  
    While ( parent > 0 AND Heap[parent] <  
        Heap[index]) do :  
        Swap(Heap[parent] , Heap[index])  
        index = parent;  
        parent = index/2;
```

Time Complexity

Method 1 -: $O(\log(\text{index}) + \log(\text{size}))$.

Method 2-: $O(\log(\text{index}))$ or $O(\log(\text{size}) - \log(\text{index}))$



Thank **you.**