15146312 park, Gangmin

Homework Assignment #4

Due: 2020/05/20 12:00

Assessment policy:

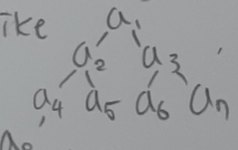
- Give full points when correct, 1/n for solving each n subproblems. 0 for totally wrong or none, -1 for each errors.
- There may be partial points for proofs if the direction is correct.

1. Understanding max-heap (6 pts)

Answer the following questions about max-heap, and explain your answer. *(3 pts each)*

(a) Is an array that is in sorted order a min-heap? Explain your answer.

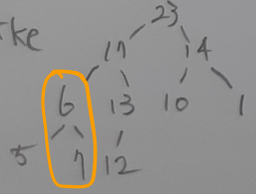
If the array is sorted in ascending order, it can be a min-heap. Because the definition of min-heap is that the parent's node should be less than child node. In the case of ascending sorted array, $\langle a_1 a_2 a_3 \dots a_n \rangle$, it can be presented by heap like



so parent's nodes of these heap are always less than child nodes.

(b) Is the array with values $\langle 23, 17, 14, 6, 13, 10, 1, 5, 7, 12 \rangle$ a max-heap? Explain why.

In the Max-heap, parent's node should be greater than child's nodes. when these array transform to heap, it is like

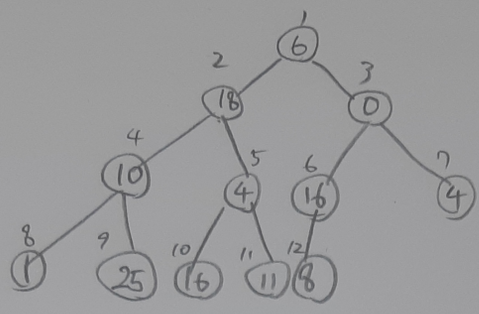


It is not max-heap because the child node 7 is greater than the parent's node 6.

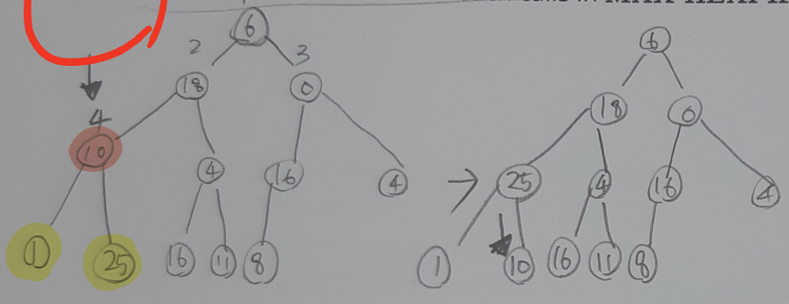
2. Applying for heapsort (12 pts)

For array $A = \langle 6, 18, 0, 10, 4, 16, 4, 1, 25, 16, 11, 8 \rangle$, illustrate each steps for heapsort. *(3 pts each)*

(a) Draw the binary tree of array A, using the model in the lecture.

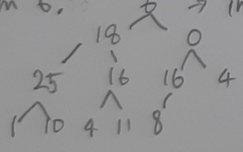


(b) Illustrate the state of A for each calls in MAX-HEAPIFY$(A, 4)$.

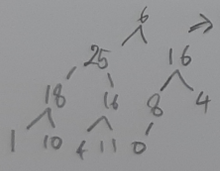


(c) Illustrate the operation of BUILD-MAX-HEAP on array A.

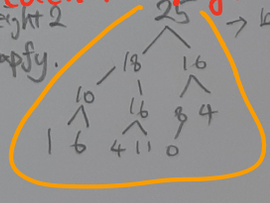
BUILD-MAX-HEAP iterate from $\lfloor n/2 \rfloor$ down to 1 running MAX-HEAPIFY. So in this problem, n is 12, it start from 6.



→ This is results of heapfy on height 1.



→ then, nodes on height 2 are changed by heapfy.



→ lastly, root node is changed by heapfy

If you are not going to illustrate for each node, you should be very clear.

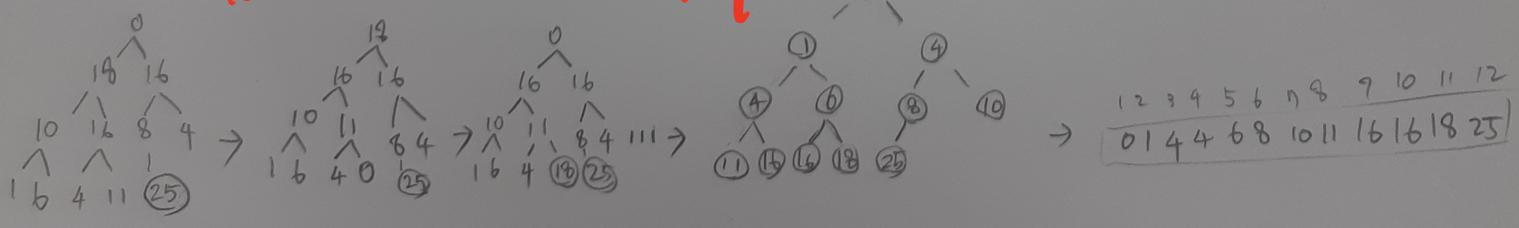
−1

(d) illustrate the operation of HEAPSORT on array A. Draw the result binary tree.

Change the root node and last node from the heap. then, run HEAPFY for root node without last node.

Repeat above process until heap became empty.

You should illustrate this. −1

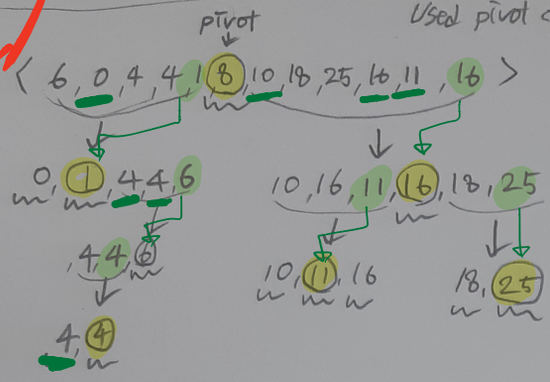


3. Applying for quicksort (13 pts)

For array $A = \langle 6, 18, 0, 10, 4, 16, 4, 1, 25, 16, 11, 8 \rangle$, illustrate each steps for quicksort.

(a) Illustrate the state of A for each calls in beginning of QUICKSORT. *(3 pts)*

Used pivot at last index.



(b) We will apply a different partition method using the median of 3. *(10 pts)* Instead of selecting the last element in the subarray as the pivot element, we select the median of the first, middle, and the last element as pivot (the middle element is computed as $\lfloor n/2 \rfloor$). Write pseudocode for MEDIAN-3-PARTITION, and illustrate the state of A for each calls using the partition method in quicksort. *(10 pts)*

```
MEDIAN-3-PARTITION(A,p,r)
mid = ⌊(p+r)/2⌋ - 1
if A[p] > A[mid]
    exchange A[p] with A[mid]
if A[p] > A[r]
    exchange A[p] with A[r]
if A[mid] > A[r]
    exchange A[mid] with A[r]
if r-p > 2
    exchange A[mid] with r-1
    x = A[r-1]
    i = p
    for j = p+1 to r-2
        if A[j] ≤ x
            i = i+1
            exchange A[i] with A[j]
    exchange A[i+1] with A[r-1]
else i = p
return i+1
```