Assignment 1a

Write pseudo code for the following functions using Min-Heap:

– MIN-HEAPIFY

Algorithm: MIN-HEAPIFY(A, i, n)

```
1. l ← LEFT(i)
2. r ← RIGHT(i)
3. if l ≤ n and A[l] < A[i]
4.     then smallest ← l
5.     else smallest ← i
6. if r ≤ n and A[r] < A[smallest].
7.     then smallest ← r
8. if smallest ≠ i
9.     then exchange A[i] ↔ A[smallest]
10.         MIN-HEAPIFY(A, smallest, n).
```

– BUILD-MIN-HEAP

Algorithm: BUILD-MIN-HEAP(A).

```
1. n = length[A]
2. for i ← ⌊n/2⌋ downto 1
3.     do MIN-HEAPIFY(A, i, n).
```

- HEAP-SORT (descending order).

Algorithm: HEAP-SORT (A).

1. BUILD-MIN-HEAP(A).
2. for i ← length [A] downto 2
3. do exchange A[1] ↔ A[i].
4. MIN-HEAPIFY (A, 1, i-1).

- MIN-HEAP-INSERT

Algorithm: MIN-HEAP-INSERT (A, key, n).

1. heap-size [A] ← n+1
2. A[n+1] ← -∞
3. HEAP-DECREASE-KEY (A, n+1, key).

- HEAP-EXTRACT-MIN

Algorithm: HEAP-EXTRACT-MIN (A, n)

1. if n < 1
2. then error "heap underflow".
3. min ← A[1]
4. A[1] ← A[n]
5. MIN-HEAPIFY (A, 1, n-1)
6. return min.

- HEAP-DECREASE-KEY

Algorithm: HEAP-DECREASE-KEY(A, i, key).

1. if key > A[i].
2. then error "new key is greater than current key"
3. A[i] ← key
4. while i > 1 and A[PARENT(i)] > A[i]
   do exchange A[i] ↔ A[PARENT(i)]
   i ← PARENT[i].

- HEAP-MINIMUM

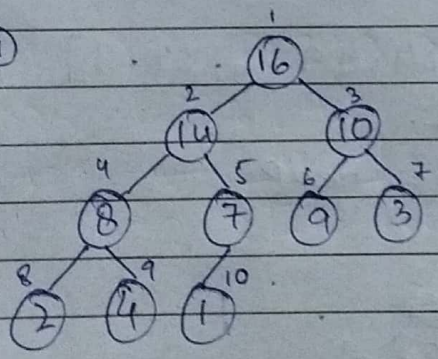
Algorithm: HEAP-MINIMUM(A).

1. return A[1].

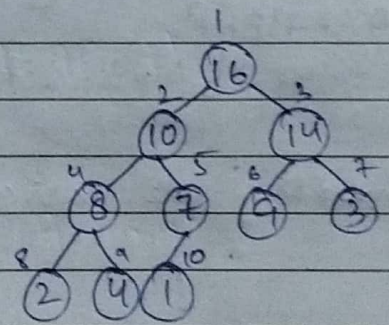
ASSIGNMENT 1b

Assuming the data in a max-heap are distinct, what are the possible locations of the second largest element?

(1)



(2)



14 is the second largest element and it can be either left or the right child of the max-heap.

Assignment 1c

(a) What is the maximum number of nodes in a max-heap of height h?

maximum number of nodes = $2^{h+1} - 1$

(b) What is the maximum number of leaves?

maximum number of leaves = $\lceil (n+1)/2 \rceil$

(c) What is the maximum number of internal nodes?

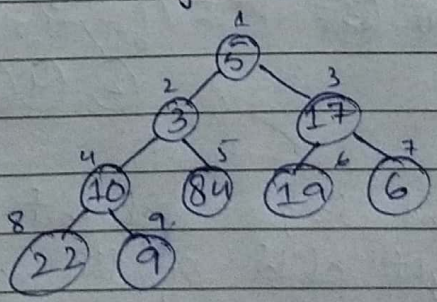
maximum number of internal nodes = $\lfloor (n-1)/2 \rfloor$

Assignment 1d

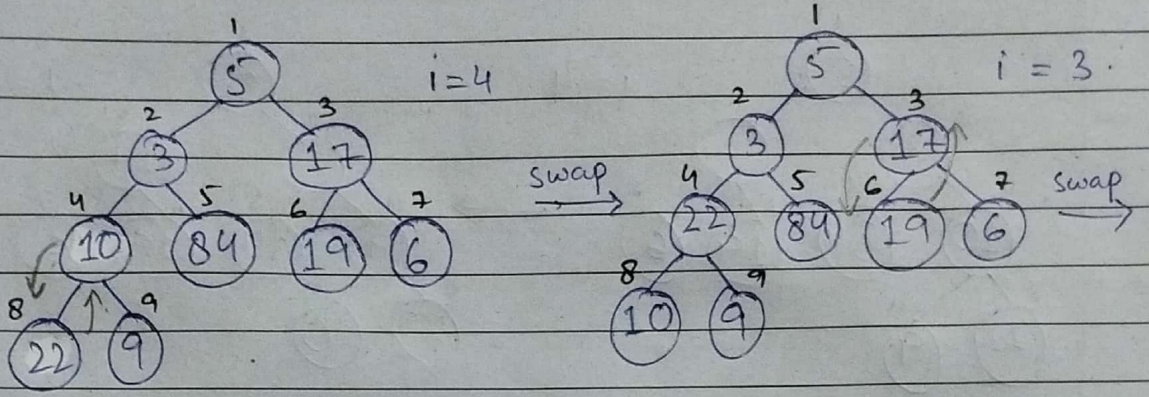
Demonstrate, step by step, the operation of Build-Heap on the array using both MIN-HEAP and MAX-HEAP.

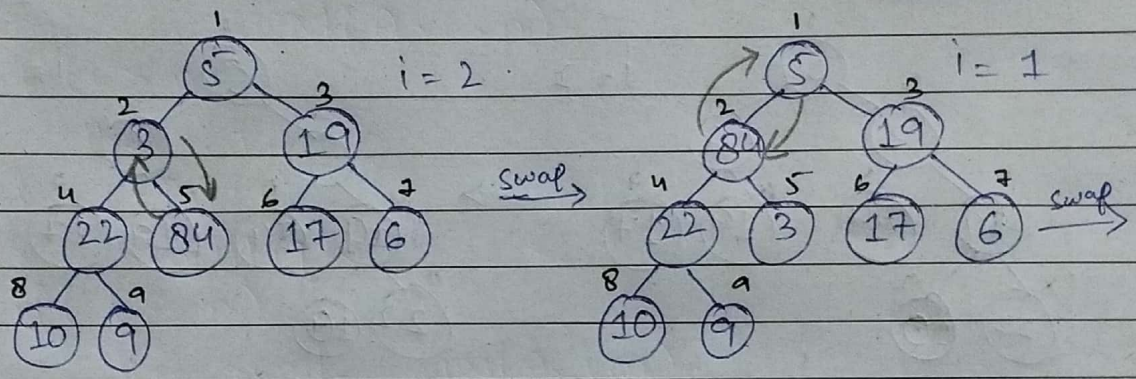
A = [5, 3, 17, 10, 84, 19, 6, 22, 9].

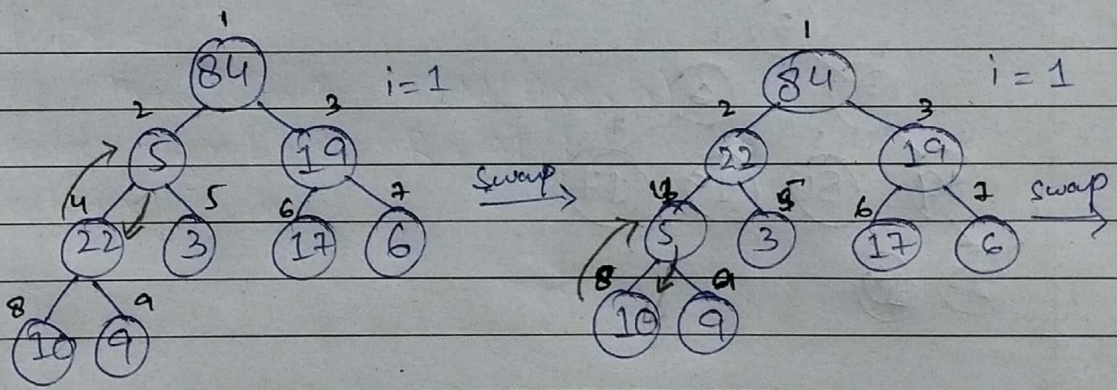
Converting the array to a heap.

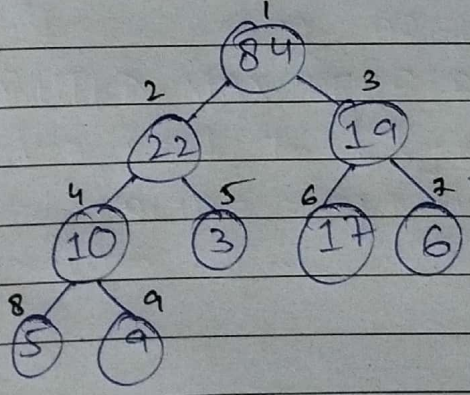


1. BUILD-MAX-HEAP $i = \lfloor n/2 \rfloor = 4$

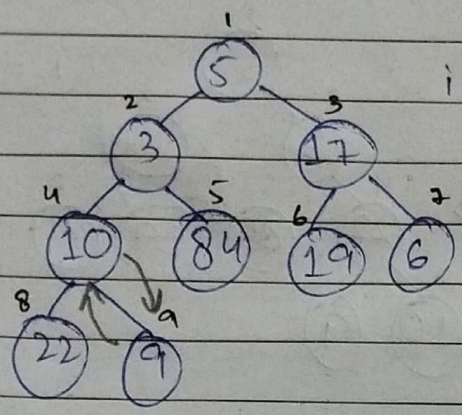


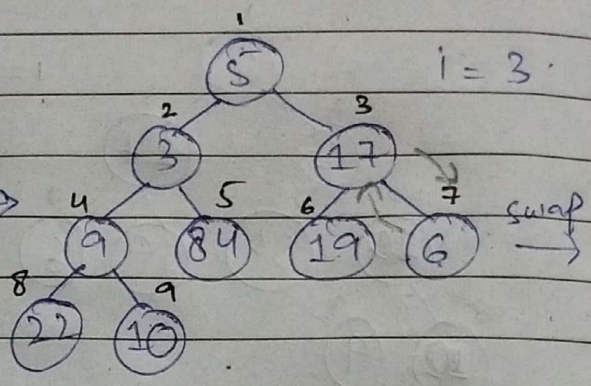


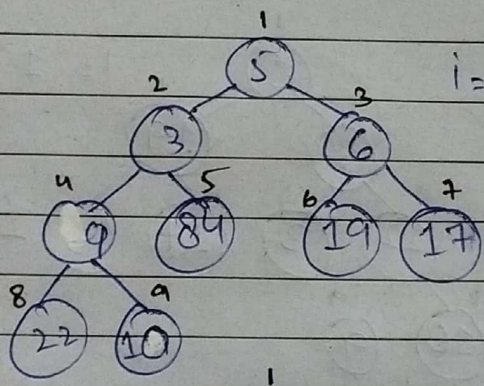


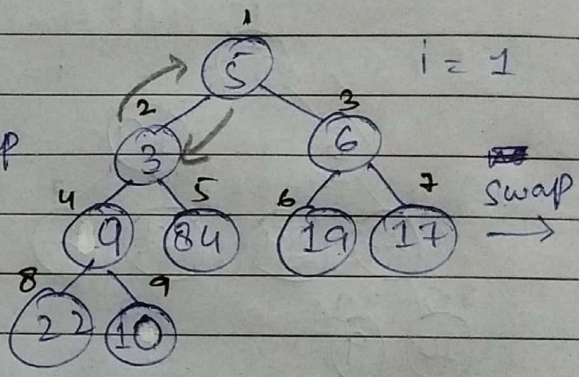


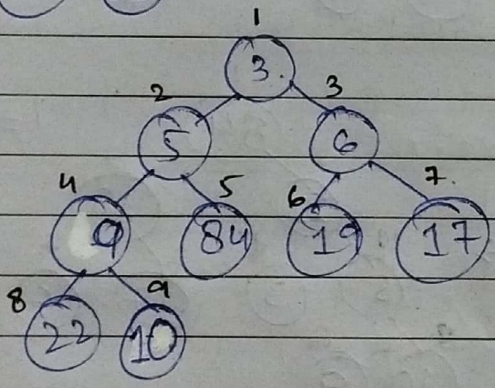
2. BUILD-MIN-HEAP $i = \lfloor n/2 \rfloor = 4$.











Assignment 1e

Let A be a heap of size n. Give the most efficient algorithm of the following tasks:

(a) Find the sum of all elements.

Algorithm: SUM(A, i, n).

```
1. if (i > n)
2.     then return 0
3. l = LEFT(i)
4. r = RIGHT(i)
5. if(l ≤ n)
6.     then sumL = SUM(A, l, n).
7. if (r ≤ n)
8.     then sumR = SUM(A, r, n).
9. sum = sumL + sumR + A[i]
10. return sum.
```

(b) Find the sum of the largest lg n elements.

Algorithm: SUM-LARGEST-LGN-ELEMENTS(A, n).

```
1. BUILD-MAX-HEAP(A)
2. for i = 1 to ⌊lg n⌋
3.     sum = sum + HEAP-EXTRACT-MAX(A, n).
4. return sum.
```