Name: (Print) PHONG VO

1. (35 points) **Heap and its application**

(1) (10 points) We have learned that the running time of MAX-HEAPIFY on a node is O(*lgn*) for a binary heap *A* that has *n* (n equals to A.length) nodes. The BUILD-MAX-HEAP algorithm below calls MAX-HEAPIFY O(n) times. This seems to suggest that the running time for BUILD-MAX-HEAP is O(*nlgn*). Check the statements in (A) to (D) and choose all correct answer(s). Briefly explain your answer. You don't need to formally approve it.

10

```
BUILD-MAX-HEAP(A)
1  A.heap-size = A.length
2  for i = ⌊A.length/2⌋ downto 1
3      MAX-HEAPIFY(A, i)
```

(A) O(*n lgn*) is an asymptotic upper bound for Build Max Heap
(B) O(*n lgn*) is a tight asymptotic upper bound for Build Max Heap
(C) O(n) is an asymptotic upper bound for Build Max Heap
(D) O(*n*) is a tight asymptotic upper bound for Build Max Heap

Build Maxheap = O(nlgn)

because O(nlgn) is a tight asymptotic upper bound of O(nlgn)

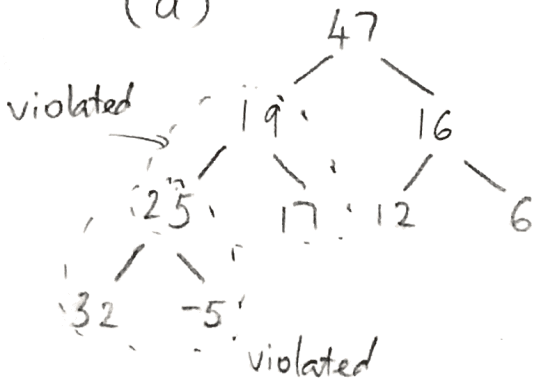
(2). (25 points) Consider the given array <47, 19, 16, 25, 17, 12, 6, 32, -5>

(a) Is this a **binary max-heap**? Justify the answer.

(b) If your answer is yes in (1), show the heap **in its binary tree view**.

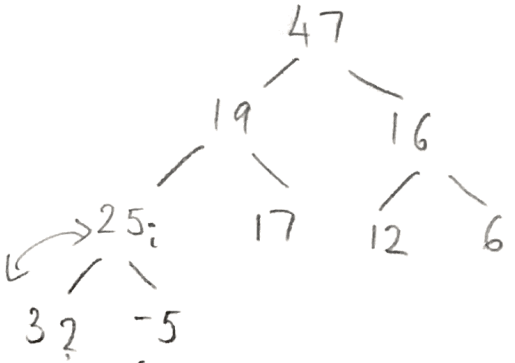
If your answer is no, make it a max-heap using the algorithm that we have learned (in the previous page). Show detailed steps of how to get the max-heap. You may use the tree view to show the changes. Show a tree for each change (a swap) in the array. Also, show the final answer (the max heap) **in an array view**.

(a)

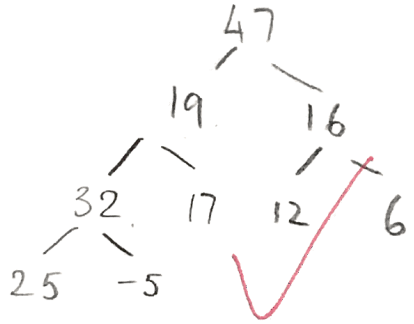


⇒ not a binary max-heap

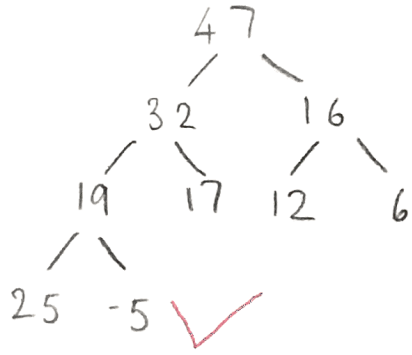
(b)



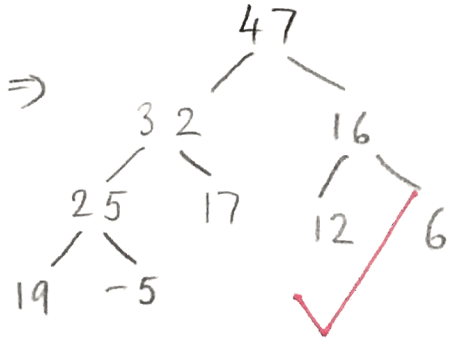
⇒



⇒



⇒

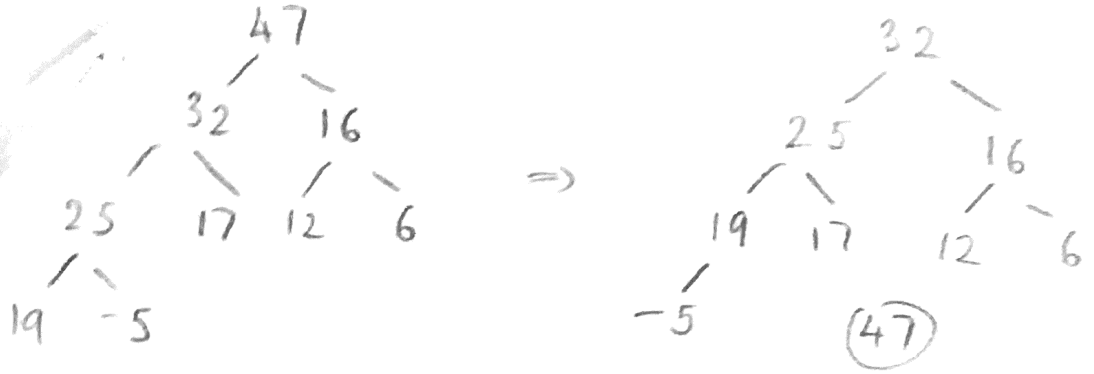


⇒ Answer : <47, 32, 16, 25, 17, 12, 6, 19, -5>
after MAX-HEAPIFY has called.

25/35

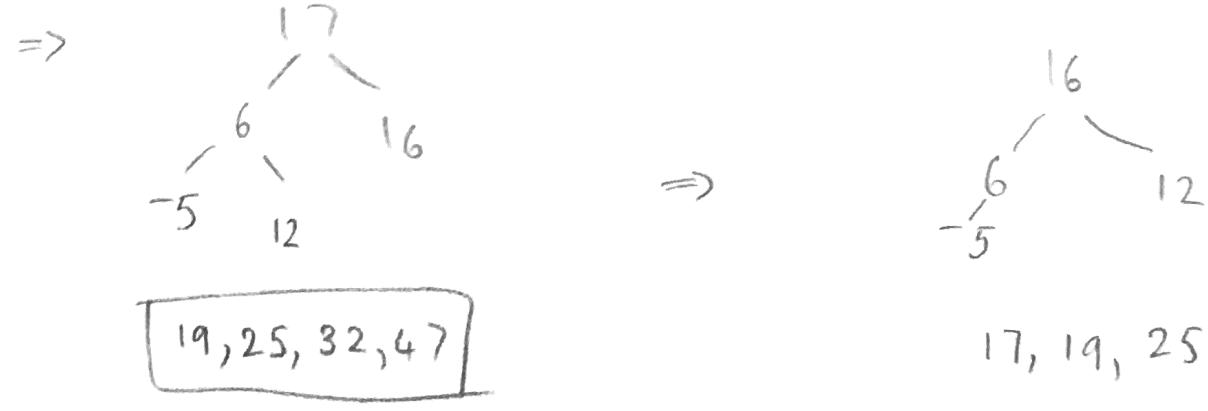
PHONG VO

Qu: 85



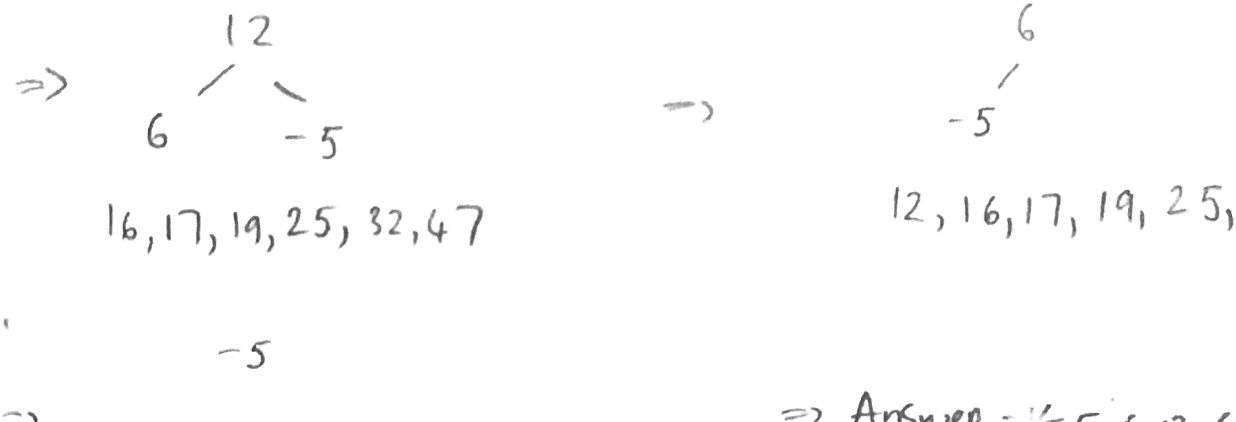
32, 47

25, 32, 47



19, 25, 32, 47

17, 19, 25, 32, 47



16, 17, 19, 25, 32, 47

12, 16, 17, 19, 25, 32, 47

-5

=>

6, 12, 16, 17, 19, 25, 32, 47

=> Answer: -5, 6, 12, 6, 17, 19, 25, 32, 47