

1. Where in a min heap the largest element resides? (Assume all elements are distinct) Explain.

Suppose that we have a min heap with n elements where $n > 1$. Suppose further that the largest element was an internal node, then that means it has atleast one child. This would be a contradiction since the largest element would be larger than its children which contradicts the min-heap property. we conclude that the largest element must be a leaf.

2. Make the min heap by successive insertions into an initially-empty min heap. Re-draw the heap each time an insertion causes one or more swaps.

$[-4, 2, 10, 12, -1, -3, 15, 76]$

Iteration tree

1.

(-4)

2.

(-4)
6

3

(-4)
6
10

4

(-4)
6
10
12

5

(-4)
6
10
12
2

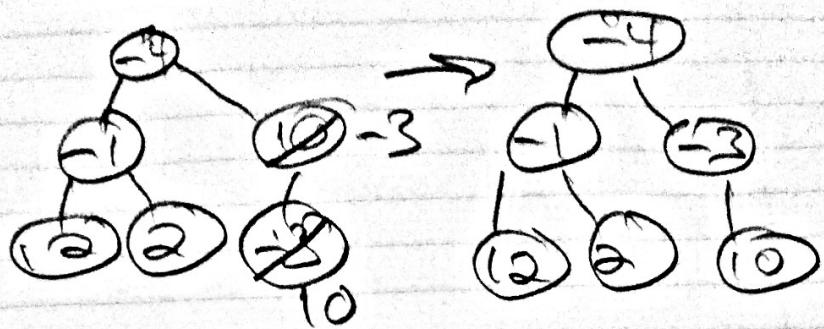


(-4)
-1
12
2
10

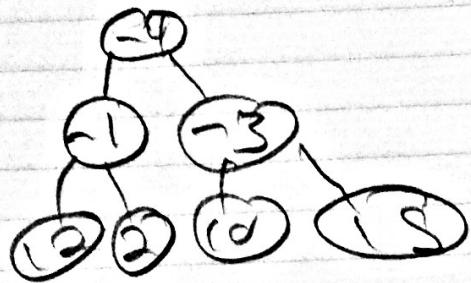
4 iteration

tree

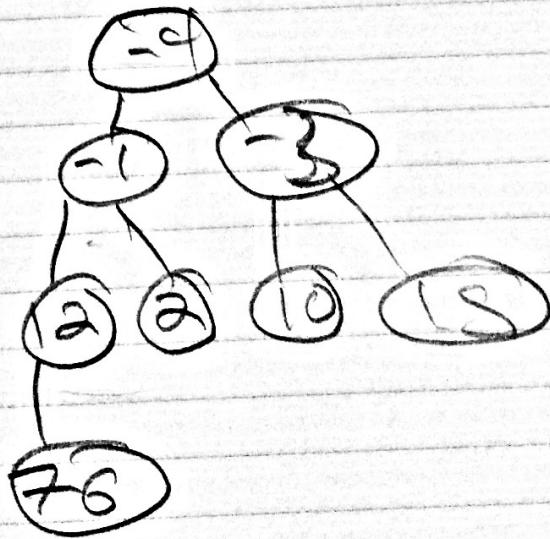
6



7

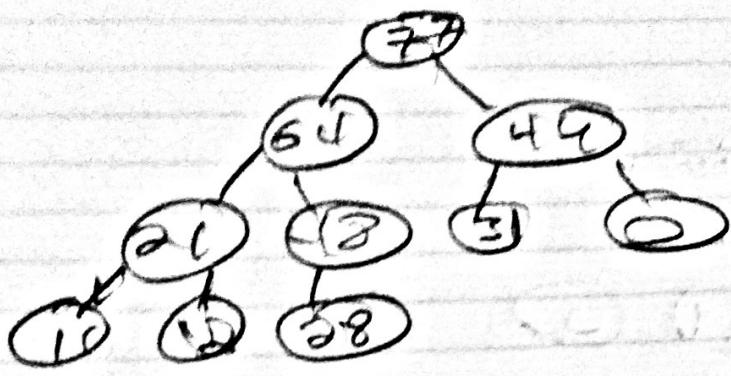


8

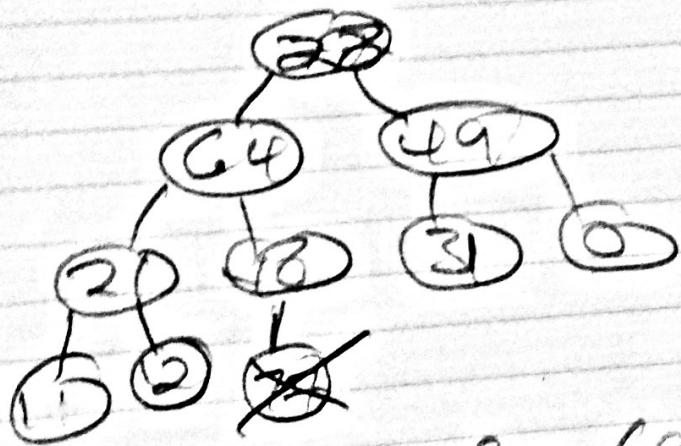


7. Delete the root of the below max-heap

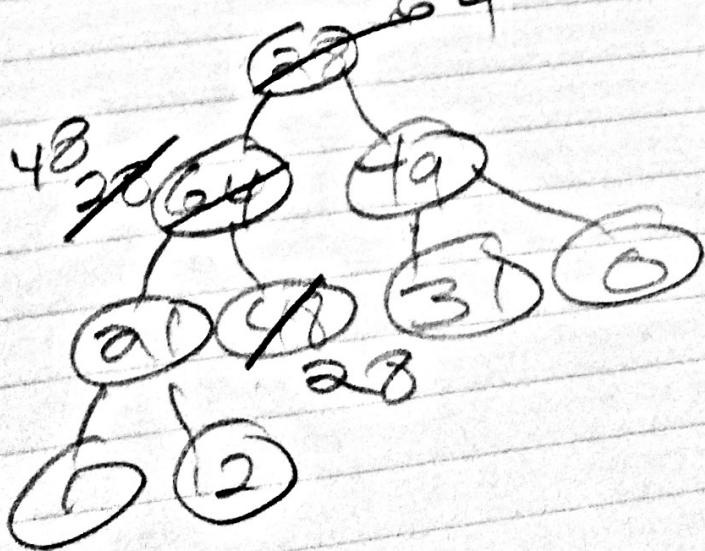
[77, 64, 49, 21, 48, 31, 0, 1, 12, 28]



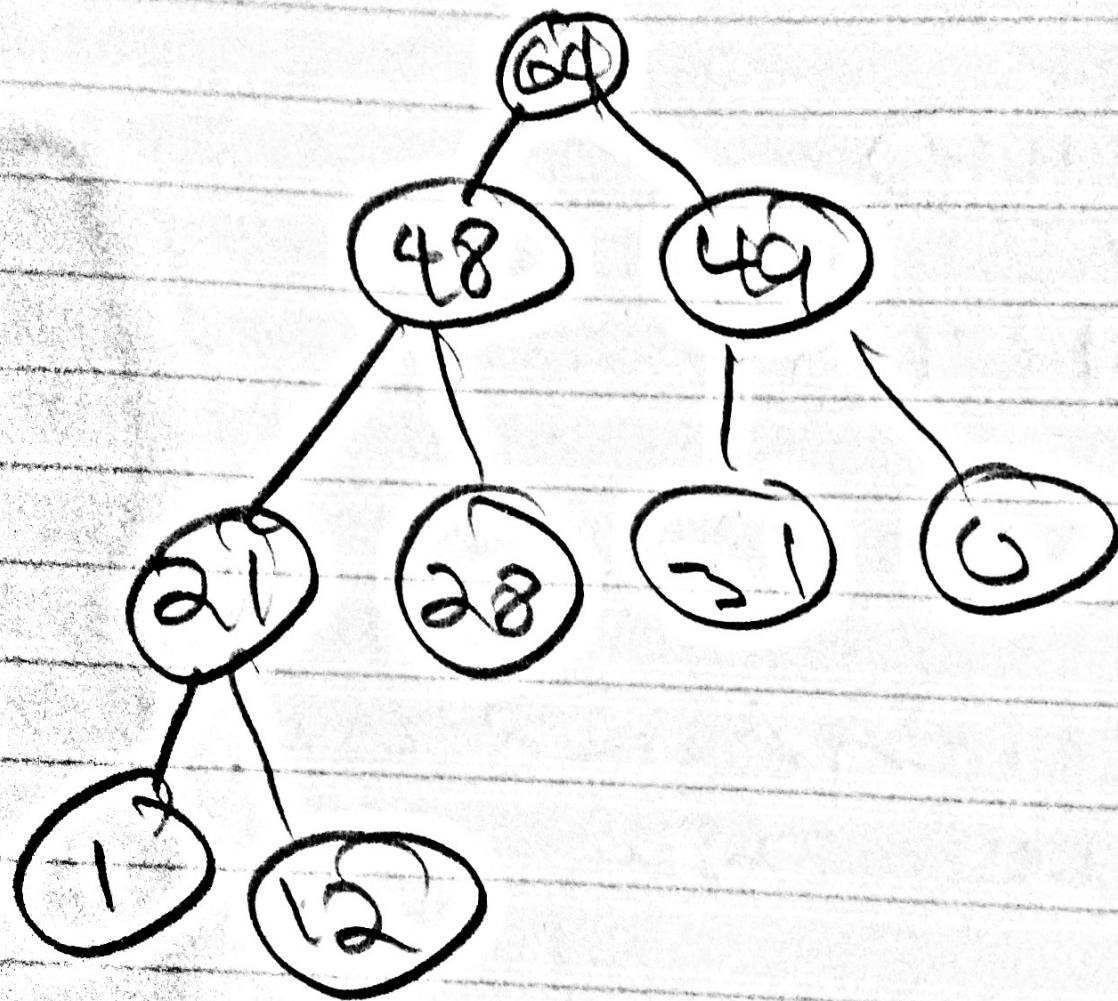
Swap 77 and 28
then reverse 72



now leafify (a) (b)

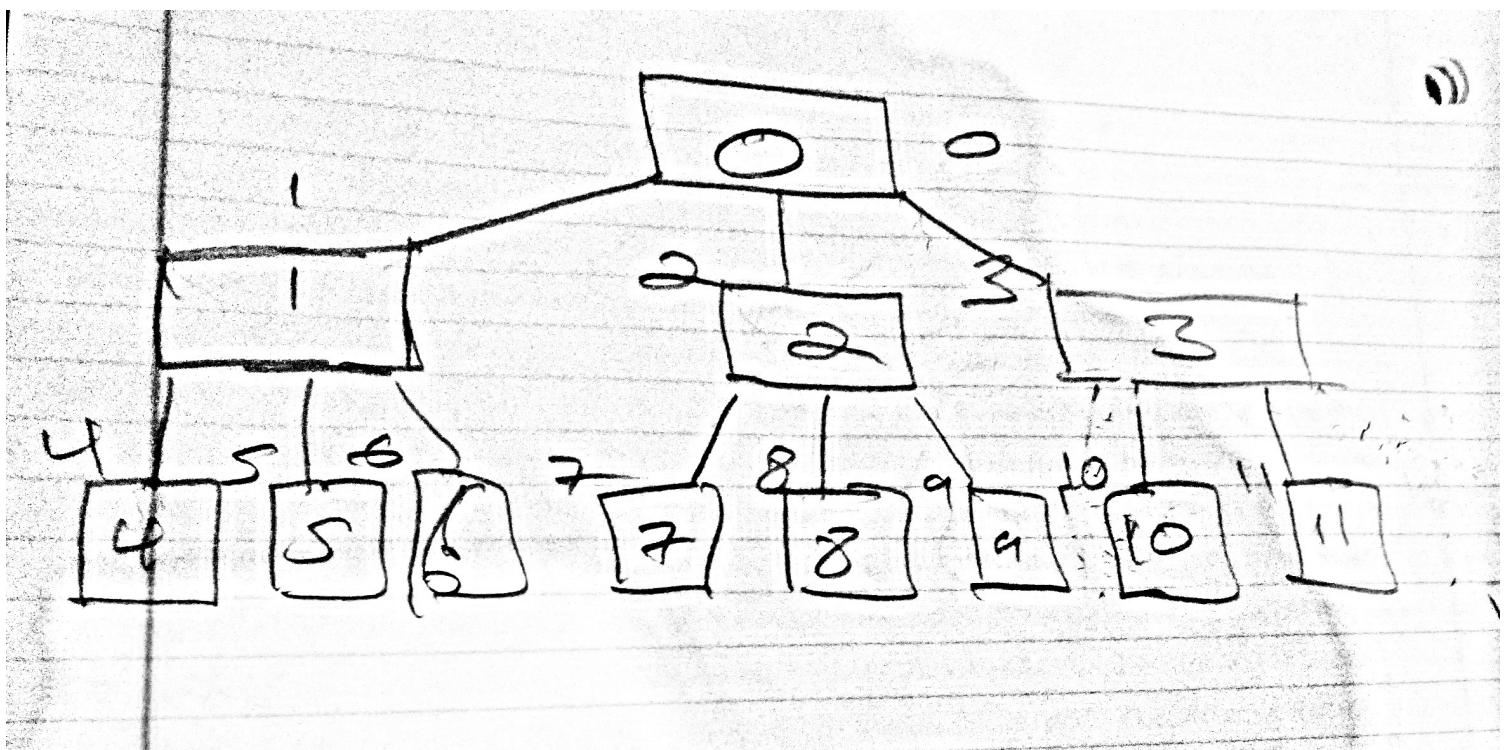


result



- 9.** Suppose that instead of binary heaps, we wanted to work with ternary heaps. Suggest an appropriate indexing scheme so that a complete tree will yield a contiguous sequence.

$$\begin{aligned}left_child &= 3 \cdot i + 1 \\middle_child &= 3 \cdot i + 2 \\right_child &= 3 \cdot i + 3 \\parent &= \lfloor \frac{i - 1}{3} \rfloor\end{aligned}$$



10. Use induction to prove that $1 + 2 + 4 + \dots + 2^h = 2^{h+1} - 1$.

Proof. We prove by induction

Let $P(h) = 1 + 2 + 4 + \dots + 2^h = 2^{h+1} - 1$

Base case: $h = 0$

$$1 = 1 = 2^{0+1} - 1.$$

Thus $P(0)$ holds.

Inductive step: Let $P(k)$ be true we show that $P(k + 1)$ is true, that is $1 + 2 + 4 + \dots + 2^k = 2^{(k+1)+1} - 1$

$$\begin{aligned} 1 + 2 + 4 + \dots + 2^k &= \sum_{i=0}^{k+1} 2^i \\ &= \sum_{i=0}^k 2^i + 2^{k+1} \\ &= 2^{k+1} - 1 + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+1+1} - 1 \\ &= 2^{(k+1)+1} - 1 \end{aligned}$$

Thus $P(k + 1)$ holds. By the principle of mathematical induction, $P(h)$ holds for all integers $h \geq 0$. \square

11. What is the minimum and maximum number of leaves in a binary heap that has height h . Explain.

When $h = 0$ then the minimum equal the maximum namely 1. Suppose that $h \geq 1$. A tree with height h must have at least one leaf at level h . The rest of the leaves are on the $h - 1$ level where the leftmost node is the only parent. Thus there are a total of $2^{h-1} - 1 + 1 = 2^{h-1}$ minimum number of leafs for height h . In a perfect tree where all levels are filled we have $\sum_{i=0}^h 2^i = 2^{h+1} - 1$ total nodes where the leafs are the last summation term and thus contribute 2^h nodes. Thus the minimum and maximum number of leaves in a binary heap that height h is 2^{h-1} and 2^h leaves for $h \geq 1$.

12. Prove that a binary heap with n elements has height $\lfloor \log_2(n) \rfloor$.

$$\begin{aligned} 2^h &\leq n & \leq 2^{h+1} - 1 \\ 2^h &\leq n & < 2^{h+1} \\ \log_2(2^h) &\leq \log_2(n) & < \log_2(2^{h+1}) \\ h &\leq \log_2(n) & < h + 1 \end{aligned}$$

which by definition of the floor $h = \lfloor \log_2(n) \rfloor$. Hence we conclude a binary heap with n elements has height $\lfloor \log_2(n) \rfloor$.

13. Prove that a binary heap with n nodes has exactly $\lceil \frac{n}{2} \rceil$ leaves.