Problem Set VI

Huy Quang Lai
132000359

Texas A&M University

27 October 2022

Problem 1

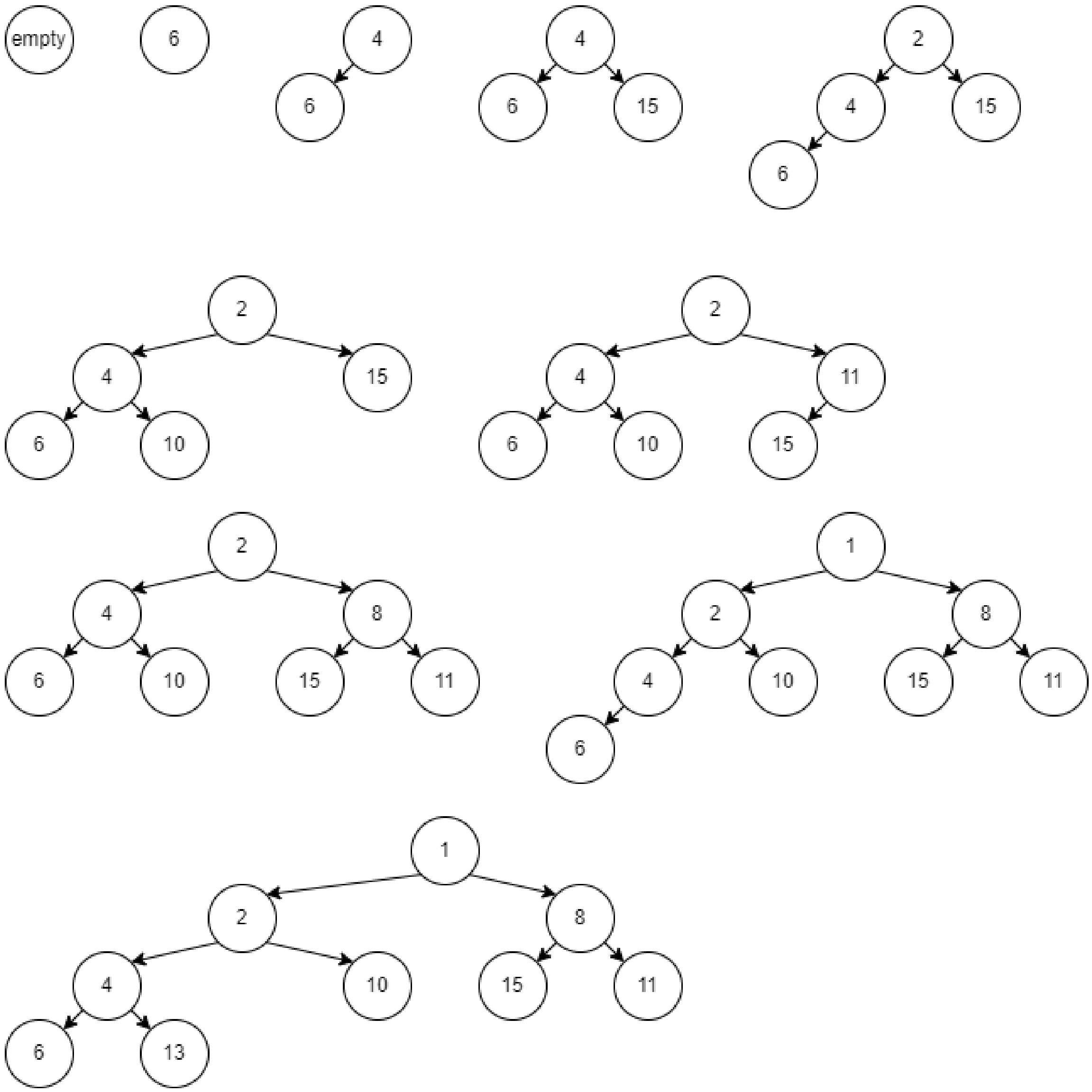

Figure 1: Insert Part 1

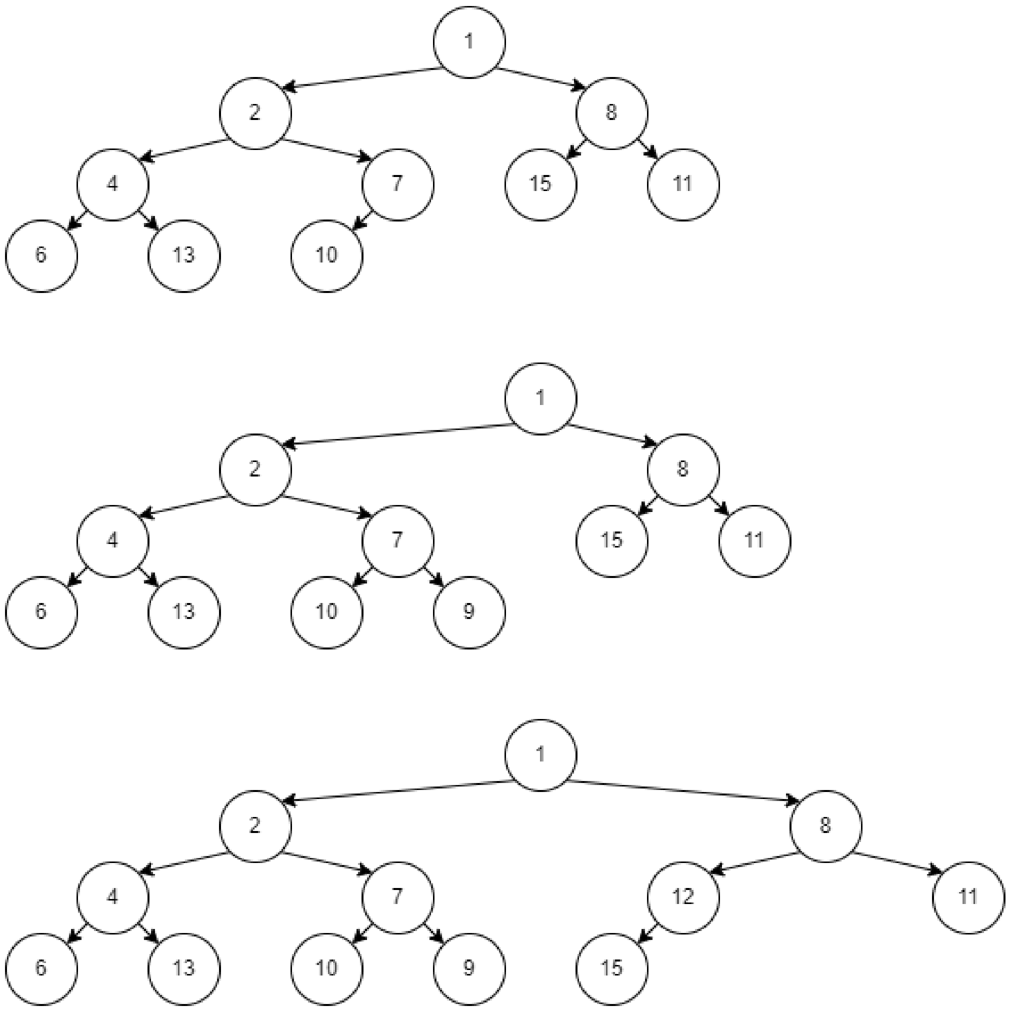

Figure 2: Insert Part 2

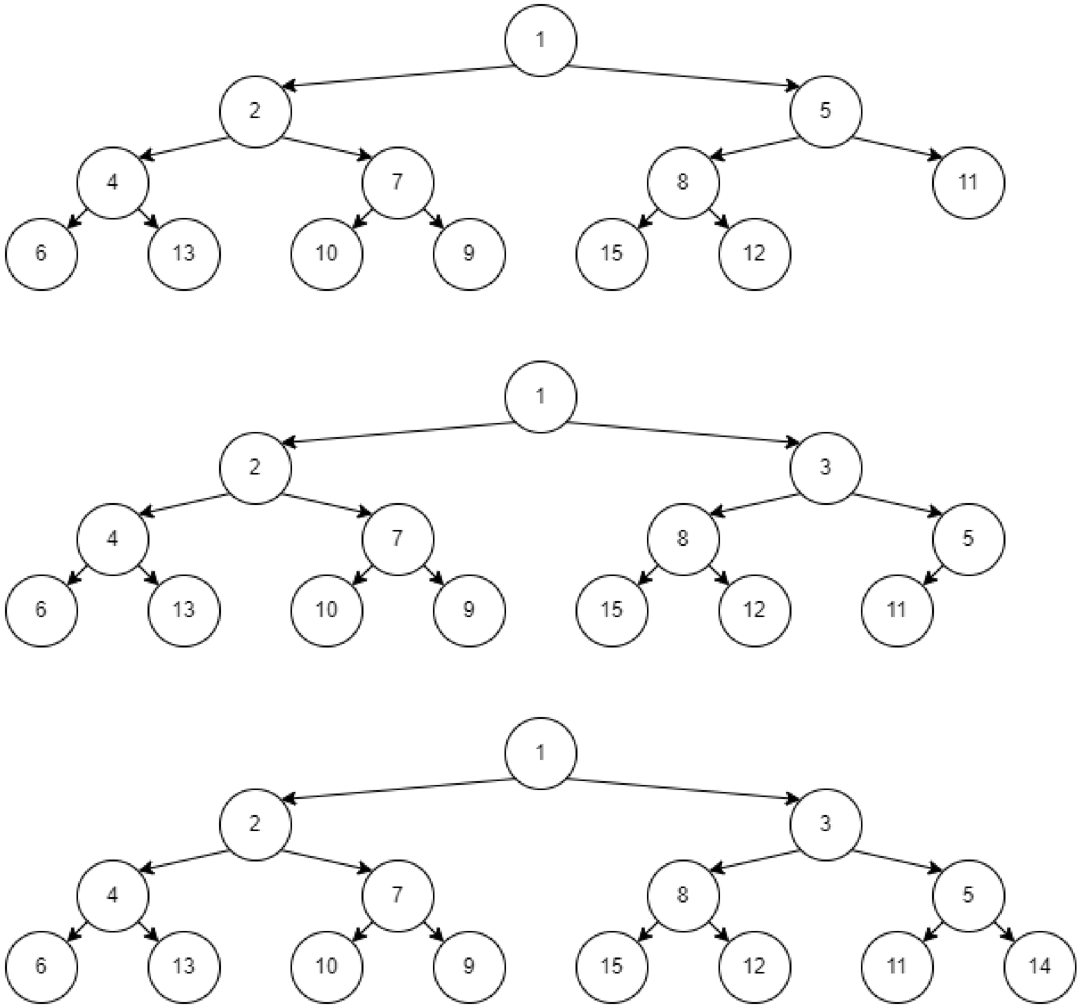

Figure 3: Insert Part 3

Problem 2

The worst cost of heapify is the sum of the "heights" of the nodes in the heap (in the binary tree sense). The sum of the heights of all the nodes in a perfect binary tree (one with all levels completely filled, the worst case for a heap) of height h is $S = \sum_{i=0}^{h} 2^i(h-i)$ because there are 2^i nodes at height $h-i$ (1 node at height h = the root, 2 nodes at height $h-1$ = the root's kids, and so on).

$$\begin{aligned}
S &= \sum_{i=0}^{h} 2^i(h-i) \\
&= \sum_{i=0}^{h} h2^i - \sum_{i=0}^{h} i2^i \\
&= h\left(\frac{2^{h+1}-1}{2-1}\right) - \sum_{i=0}^{h} i2^i \\
&= h\left(2^{h+1}-1\right) - \sum_{i=0}^{h} i2^i \\
&= h\left(2^{h+1}-1\right) - \left(2+(h-1)2^{h+1}\right) \\
&= h2^{h+1} - h - 2 - h2^{h+1} + 2^{h+1} \\
&= 2^{h+1} - h - 2
\end{aligned}$$

The number of nodes in a perfect binary tree is $N = 2^{h+1} - 1$
$\therefore S = 2^{h+1} - h - 2 = N - \log(N-1) - 2$

Problem 3

A. The first level of the tree should have n nodes, the second level of the tree should have $2n$ nodes. Therefore, slightly unbalanced tree should have $n + n + 2n$ nodes of $4n$ nodes.
The array must be at least $4n$

B. The deepest node if a height $2 \log n = \log n^2$. Since the height of a node of a binary tree is $\log x$, we can get $\log x = \log n^2$. Solving for x, we get $x = n^2$.
The array must be at least n^2

C. Same math as part B.
The array must be at least $n^{4.1}$

D. The worst case for a Binary Tree is a skew tree.
The array must be at least 2^n

Problem 4

Let the array be `{A, B, C, D, E, F, G, H}`, and we know nothing about their relative ordering.

The first four comparisons would be $A > B, C > D, E > F, G > H$.
The next two comparisons would be $A > C, E > G$.
Without loss of generality we can check $A > E$.
Finally we check if $B > C$