1

Separate Chaining

0	
1	4371
2	
3	1323, 6173
4	4344
5	
6	
7	
8	
9	4199, 9679, 1989

Linear Probing

0	9679
1	4371
2	1989
3	1323
4	6173
5	4344
6	
7	
8	
9	4199

Quadratic Probing

0	9679
1	4371
2	
3	1323
4	6173
5	4344
6	
7	
8	1989
9	4199

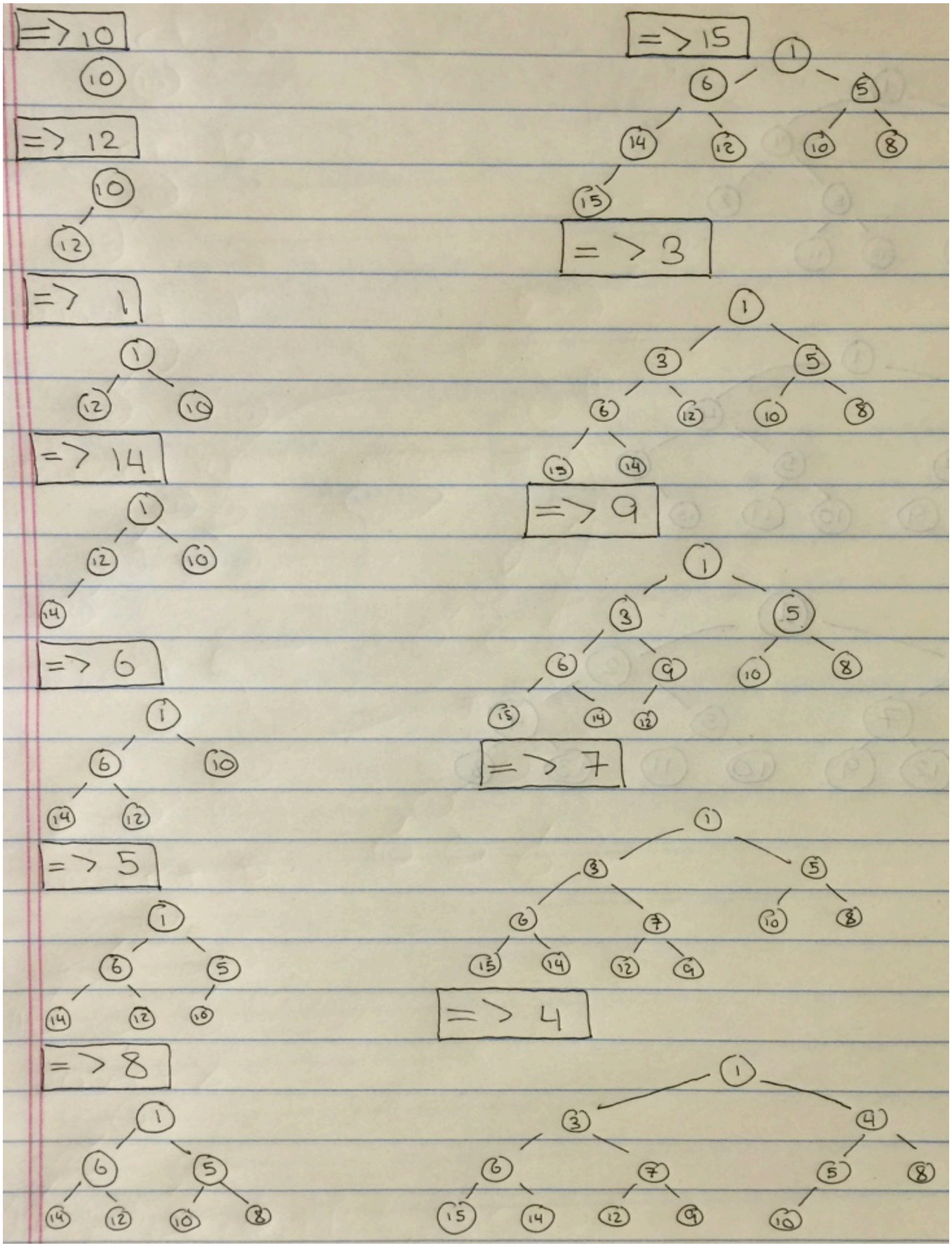
2^{nd} Hash Function

0	
1	4371
2	
3	1323
4	6173
5	9679
6	
7	4344
8	
9	4199

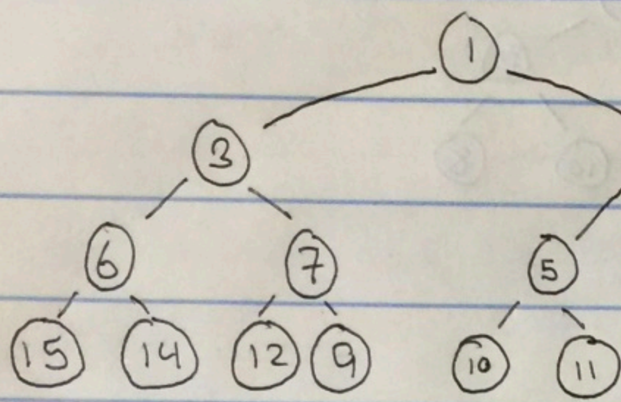
2.
Separate Chaining, Linear Probing, Quadratic Probing, 2^{nd} Hash Function
All have this table:

0	
1	4371
2	
3	
4	
5	
6	
7	
8	
9	6173
10	
11	1989
12	1323
13	4199
14	
15	
16	
17	
18	
19	9679
20	4344
21	
22	

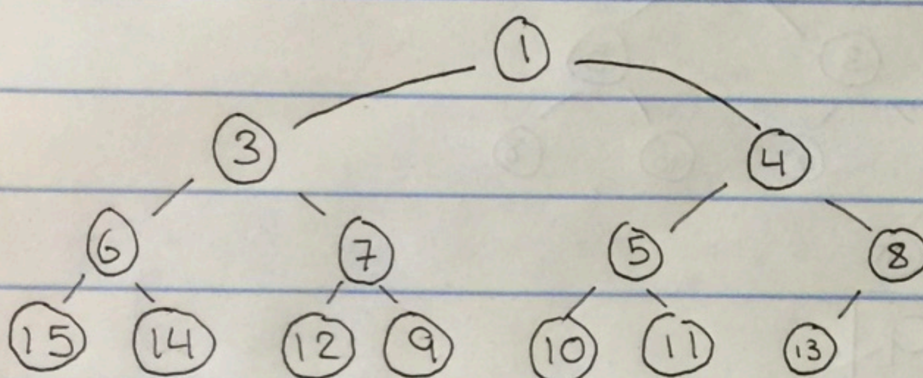
3.



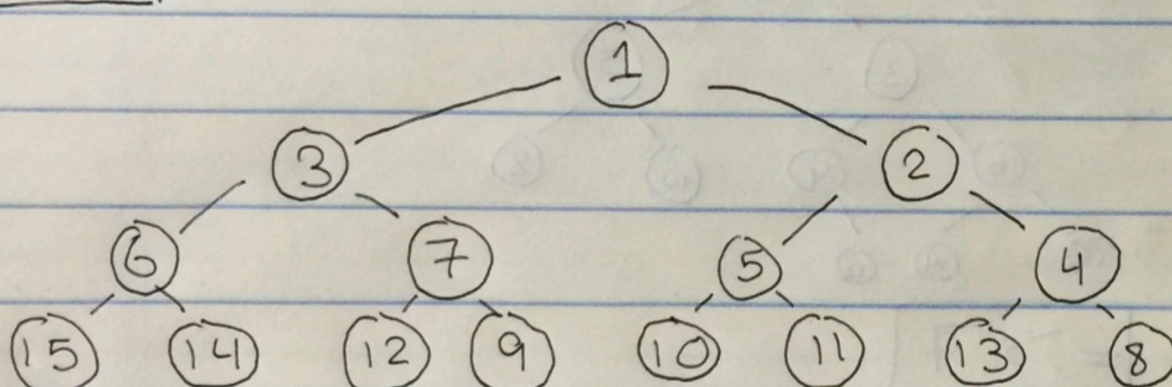
=> 11



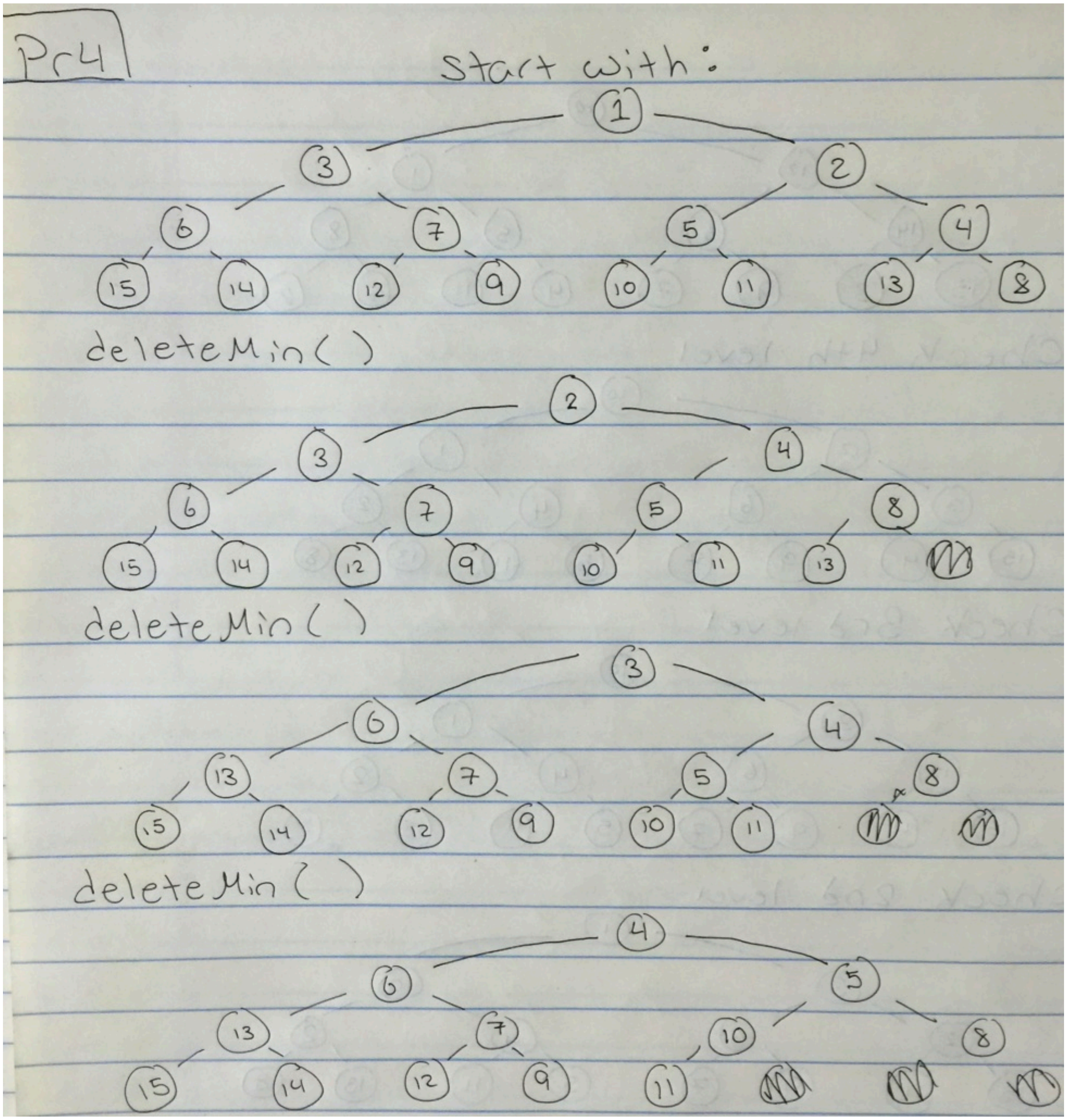
=> 13



=> 2



4.



5.

a. The heap-order property requires that every node be smaller than its parent. This means the maximum value node cannot have any children, b/c it would be bigger than them. A node with no children is a leaf.

b. At any given height(h) there are $N_h = 2^h$ nodes

$N = \sum_{h=0}^{h} 2^h = 2^{h+1} - 1$ nodes total for a complete tree of height h.

$$N = 2^{h+1} - 1$$

$$N + 1 = 2^{h+1} \Rightarrow \ln(N+1) = (h+1)\ln(2)$$

$$\hookrightarrow \frac{\ln(N+1)}{\ln(2)} - 1 = h \rightarrow \log_2(N+1) - 1 = h$$

$$\rightarrow \log_2(N+1) - \log_2(2) = h \rightarrow h = \log_2\left(\frac{N+1}{2}\right)$$

Now the # of leaves is still 2^h

$$L = 2^h = 2^{\log_2\left(\frac{N+1}{2}\right)} = \frac{N+1}{2} = \frac{N}{2} + \cancel{\frac{1}{2}} = \left\lceil \frac{N}{2} \right\rceil \quad \text{Q.E.D}$$

c. As we have already proven that the maximum node must be a leaf, and heaps do not organize themselves in any other way than the heap-order property. Hence, we must check every leaf node to find the maximum node.