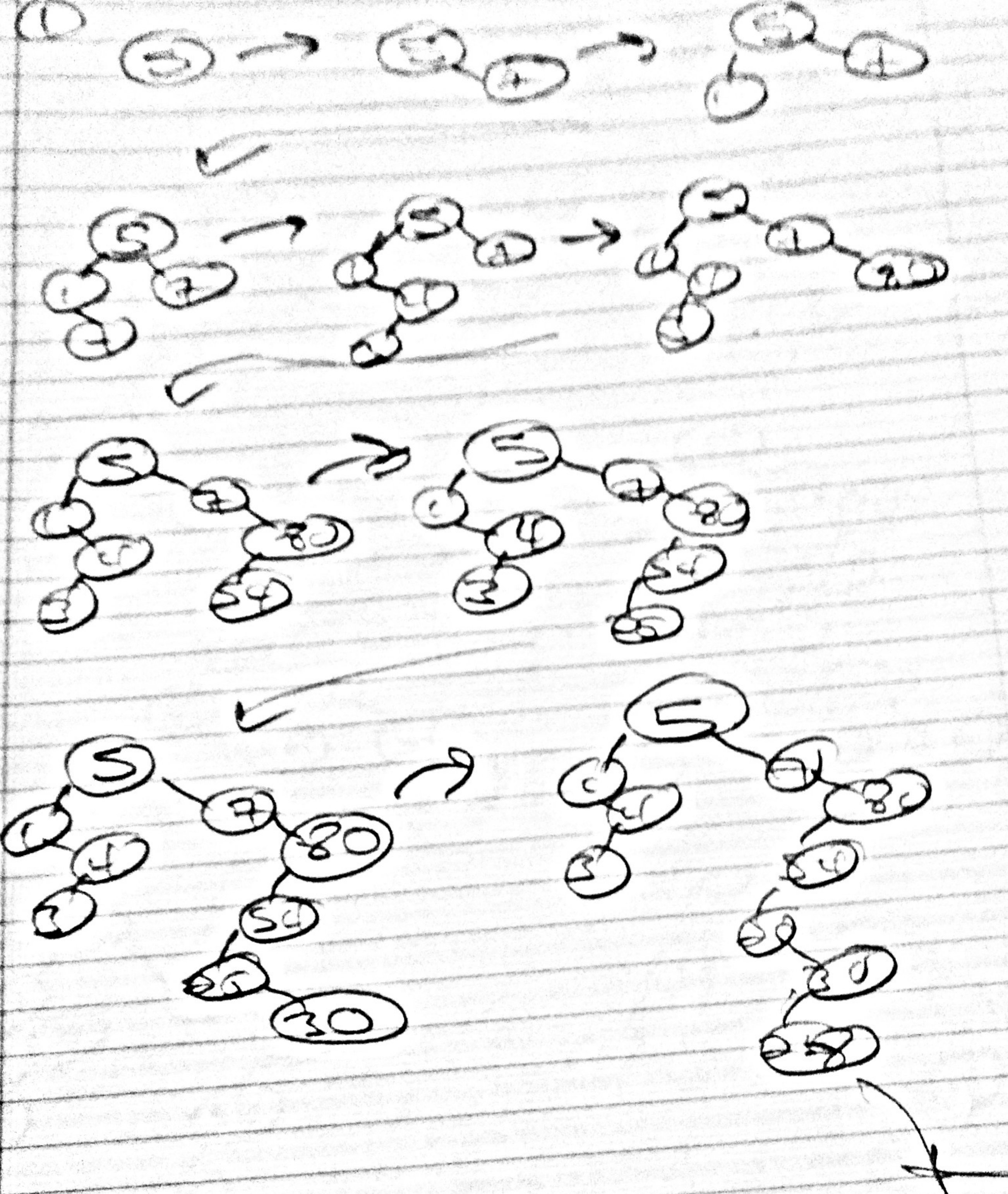
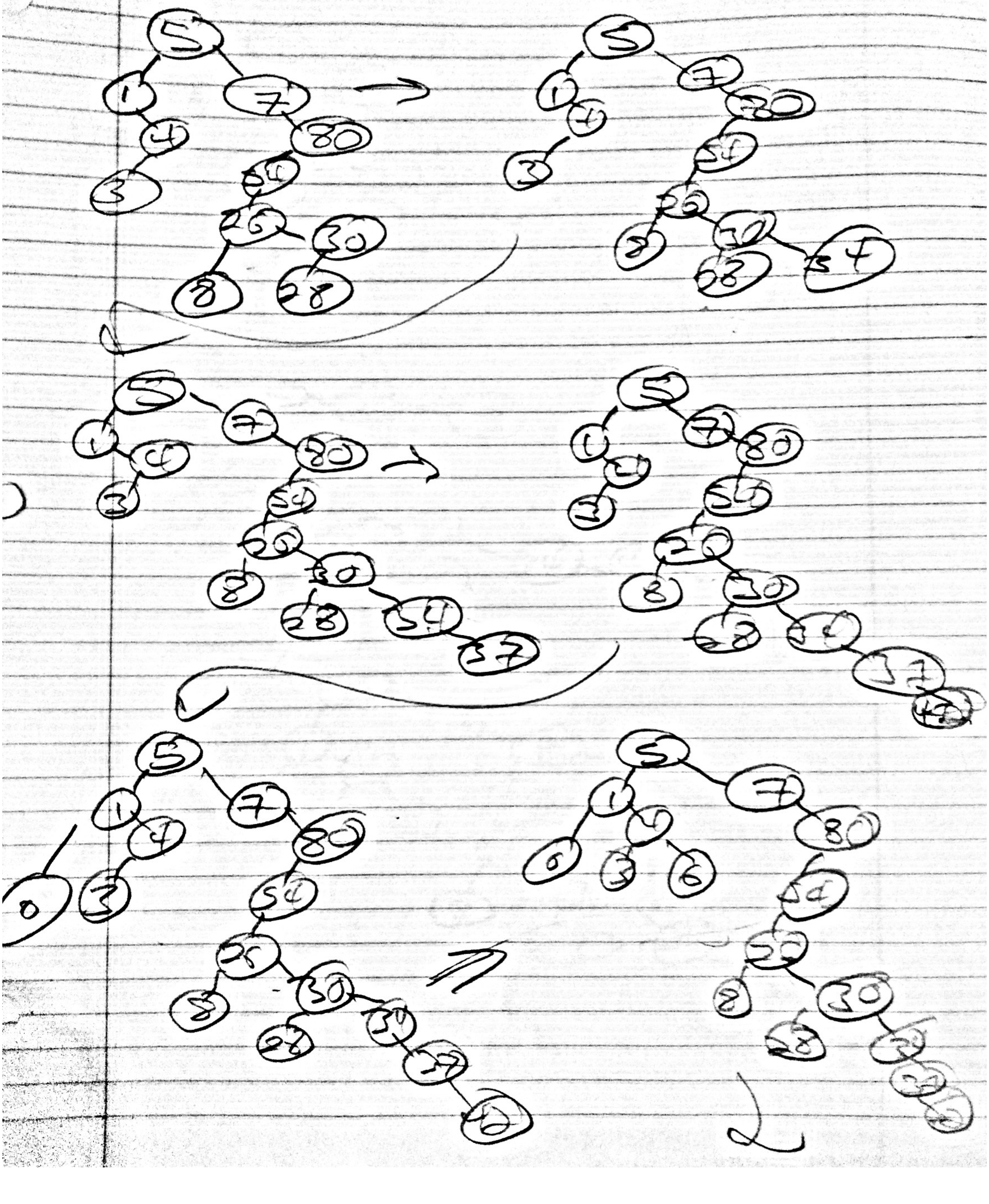
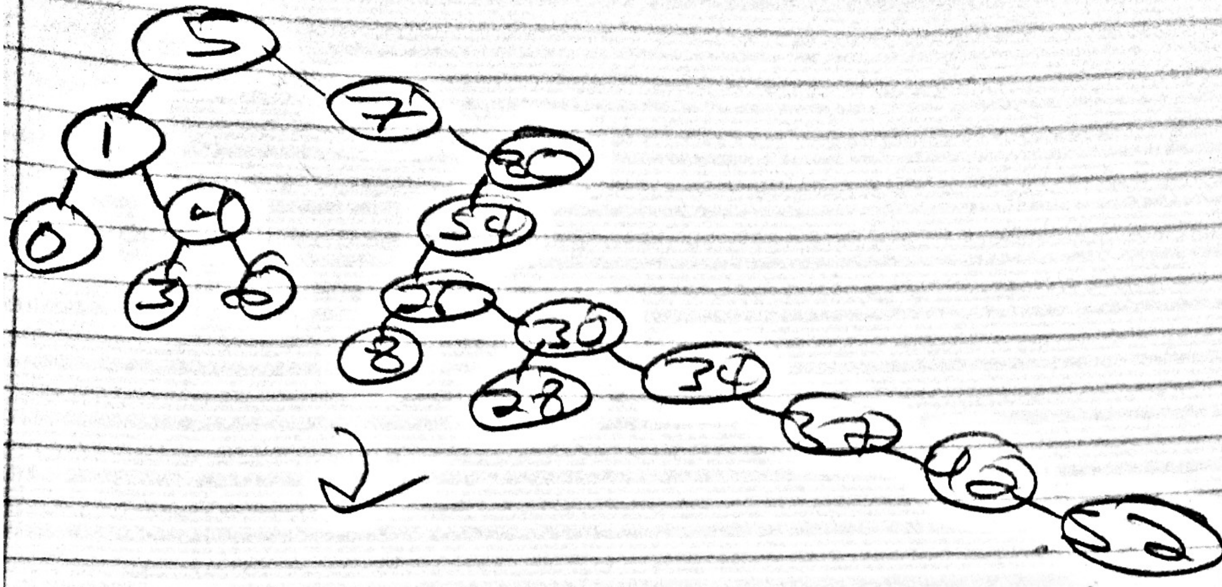
1. Make a binary search tree using these numbers

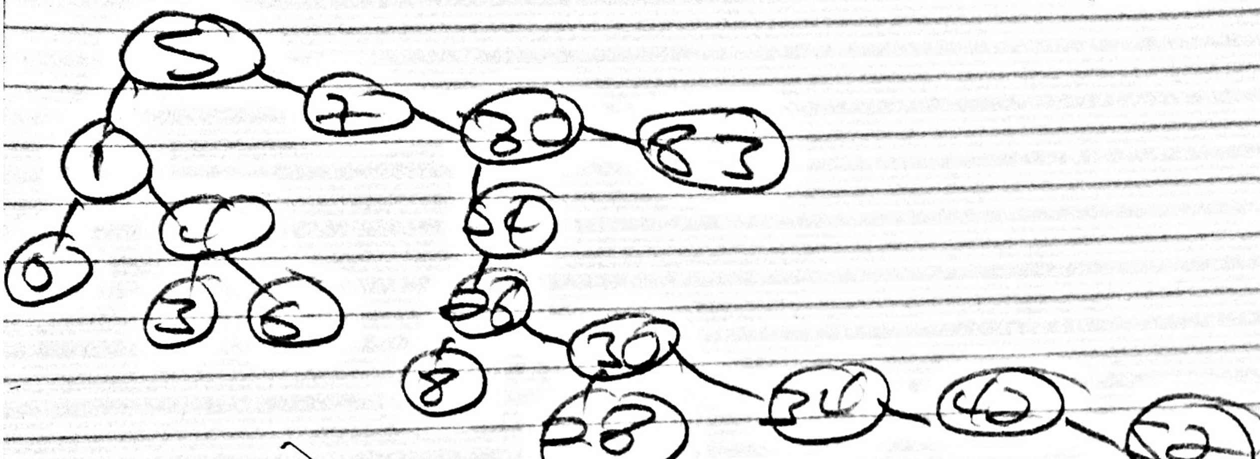
$$5, 7, 1, 4, 3, 80, 54, 26, 30, 28, 8, 34, 37, 42, 0, 6, 52, 83, 81$$

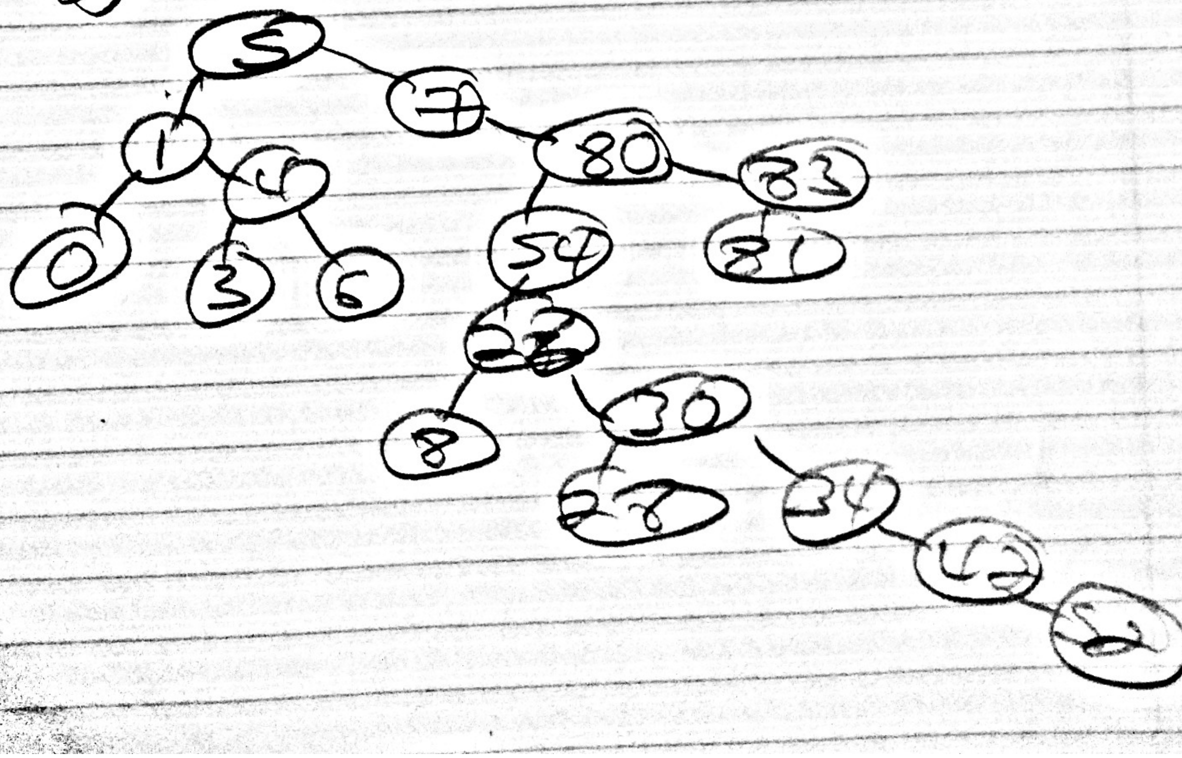
1






5
7
1
80
0
4
54
3
6
8
30
28
34
40
52


5
7
80
83
1
54
0
4
3
6
30
8
28
34
40
52


5
7
1
80
83
0
4
54
81
3
6
30
8
28
34
40
52

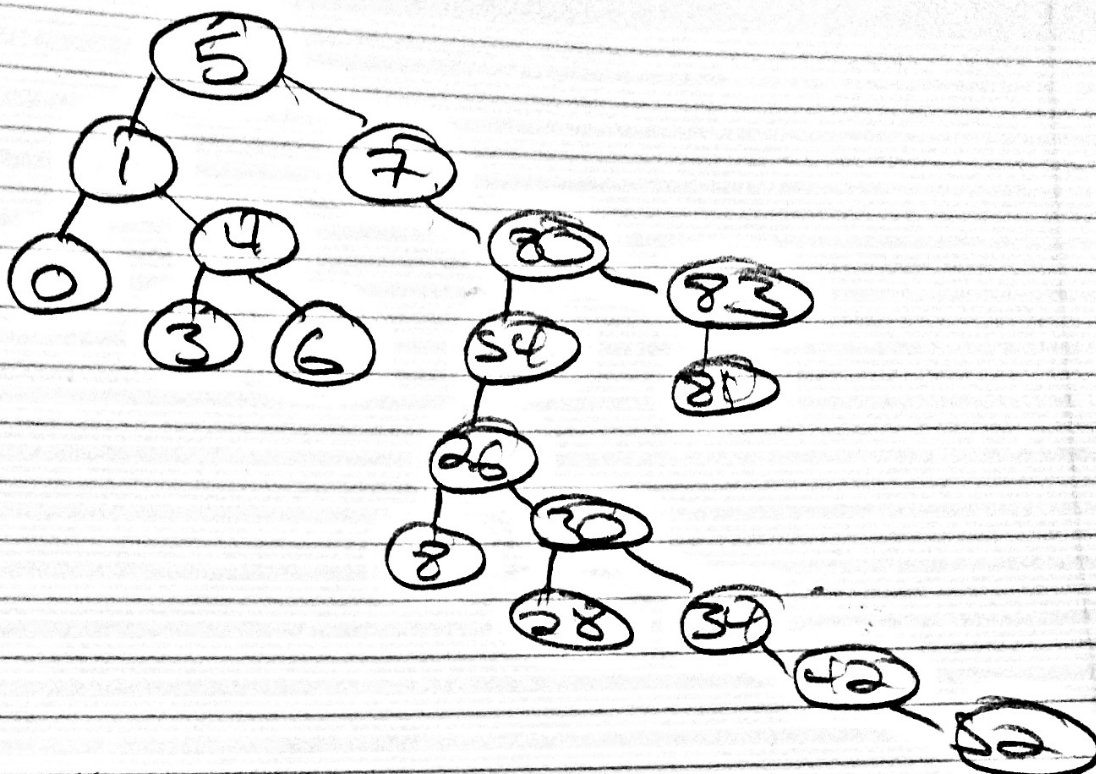
2. Delete the below nodes in your binary search tree

1. 52
2. 34
3. 37
4. 6

3. Find the successor of the below numbers in the BST in question 1.

1. 7
2. 52
3. 34
4. 30
5. 28
6. 83

3



a. Successor(7) = 8

b. Successor(52) = 54

c. Successor(34) = 42

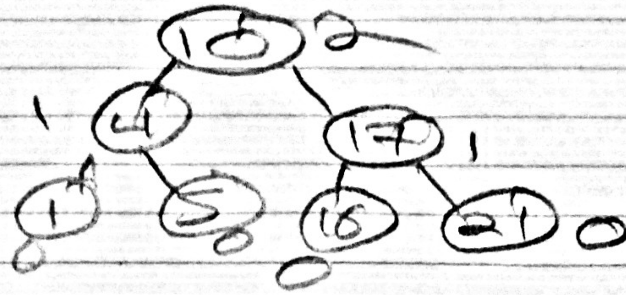
d. Successor(30) = 34

e. Successor(28) = 30

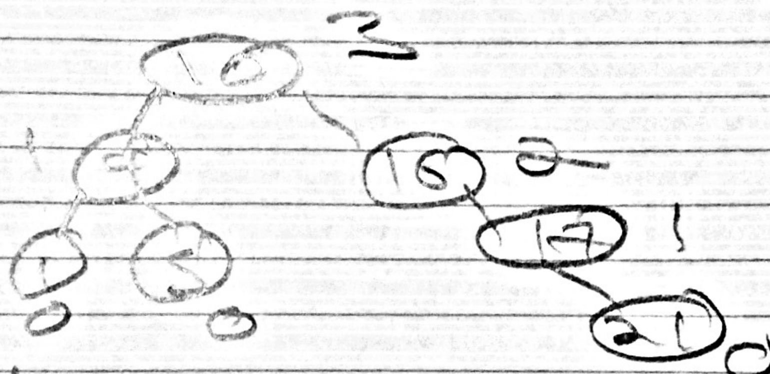
f. Successor(83) = none

4. Suppose a binary search tree to hold keys 1,4,5,10,16,17,21. Draw possible binary search trees for these keys, and having heights 2,3,4,5 and 6.

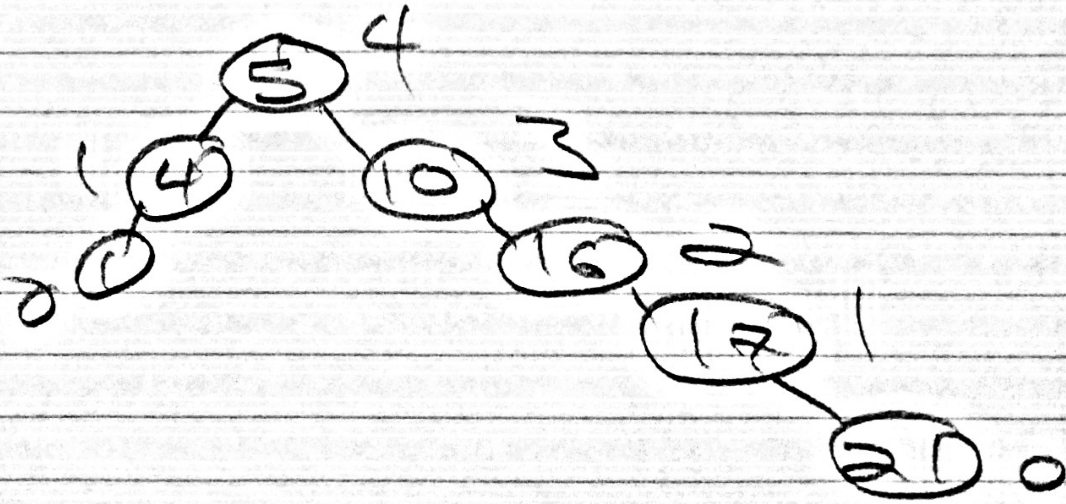
height 2



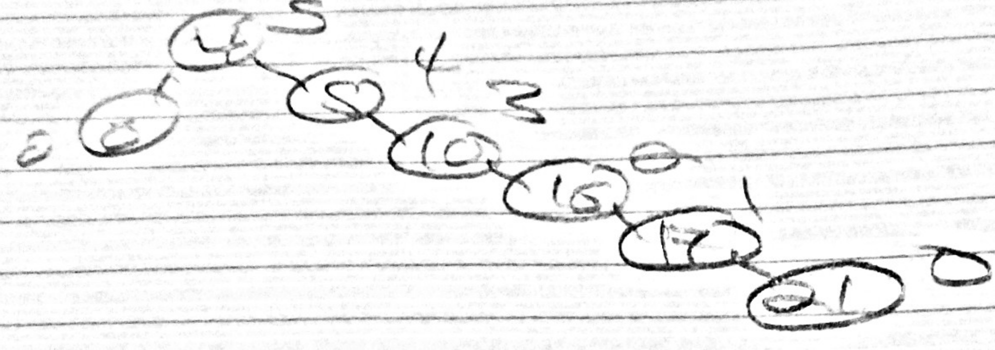
height 3



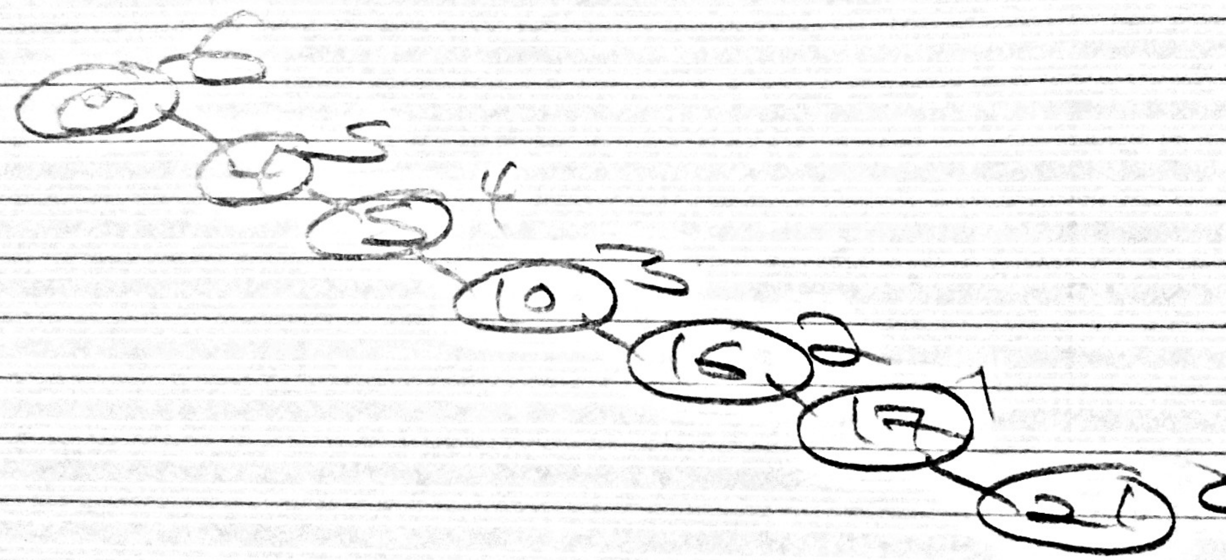
height 4



height 5



height 6

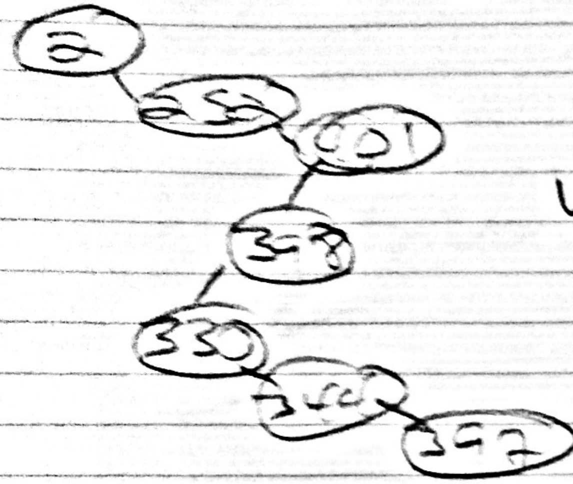


5. Suppose we have a BST that stores keys between 1 and 1000. If we perform a find on key 363, then which of the following sequences could not be the sequence of nodes examined when performing using the find() method.

1. 2,252,401,398,330,344,397
2. 924,220,911,244,898,258,362,363
3. 925,202,911,240,912,245,363
4. 2,399,387,219,266,382,278,363
5. 935,278,347,621,299,392,358,363

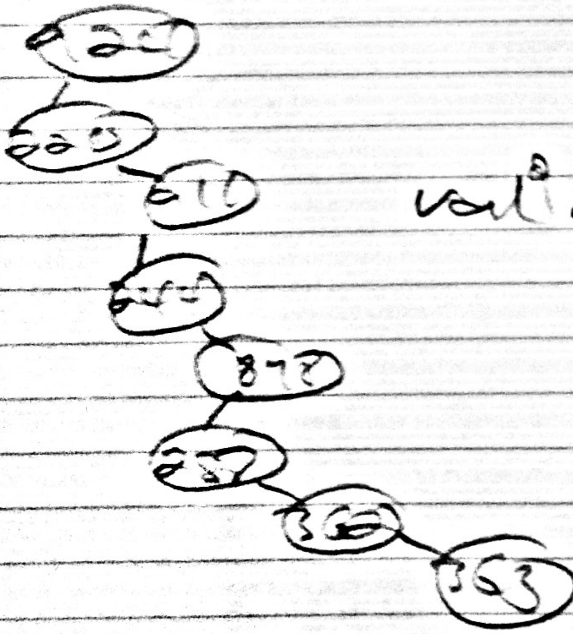
(5)

(a) 2, 252, 401, 398, 330, 344, 397



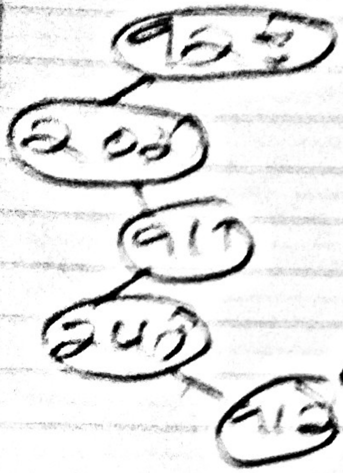
valid

(b) 924, 220, 911, 244, 898, 258, 362, 363



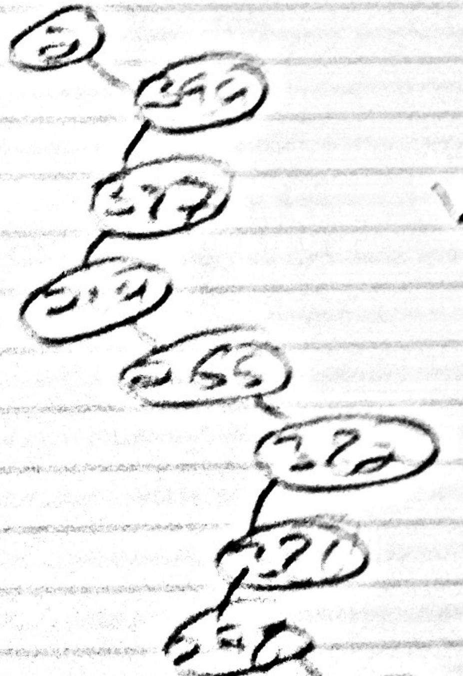
valid

c) 925, 202, 911, 240, 912, 245, 363



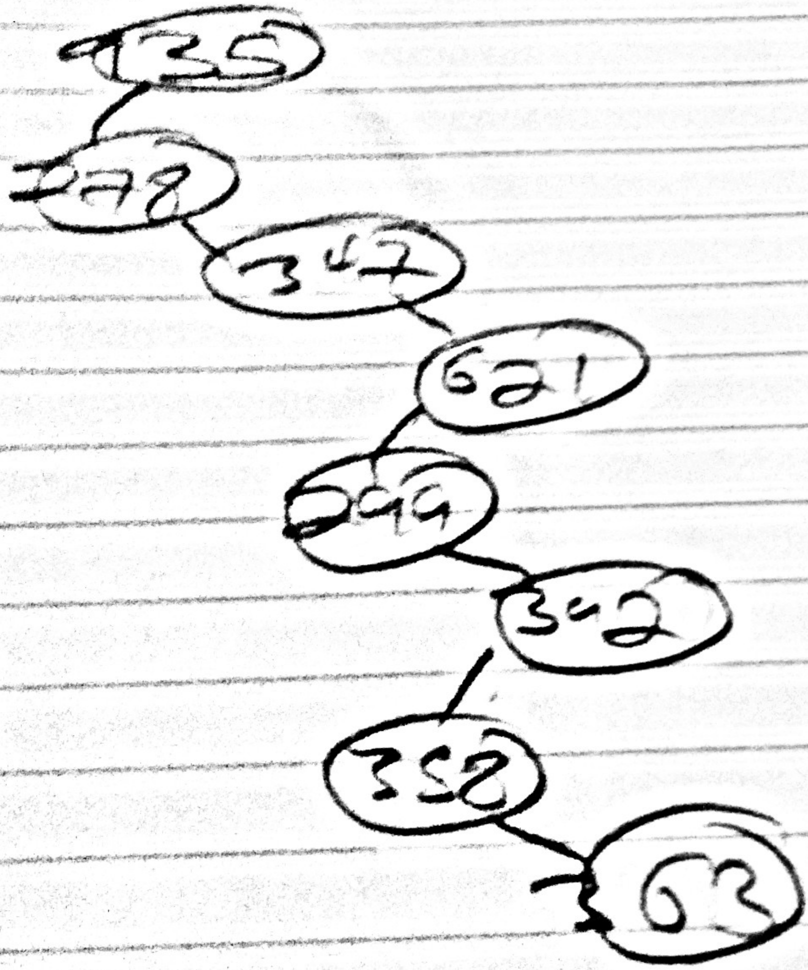
Invalid!

d) 2, 399, 387, 219, 266, 382, 381, 278, 363



Valid

(c) 935, 278, 347, 621, 299, 392, 358, 363



valid

6. Suppose a BST is constructed by repeatedly inserting distinct keys into the tree. If the number of nodes when searching for a key is k, what will be the number of nodes examined when inserting that key.

If the number of nodes that are examined for a successful search for a key is k. Then the numbers of nodes examined to find the respective node position would have been $k-1$.

7. Prove that the maximum number of nodes in a binary tree with height h is $2^{h+1}-1$.

When the tree is a perfect binary tree every level is filled and we have the summation $\sum_{i=0}^{h} 2^i = 2^{h+1}-1$. We conclude the maximum number of nodes in binary tree with height h is $2^{h+1}-1$.

8. Prove that it takes $\Omega(n \cdot log(n))$ steps in the best case to build a binary search tree having n distinct keys.

In the best case the binary search tree insertion takes $log(k)$ where k is the current number of elements in the tree. we have the summation $\sum_{i=1}^{n} log(i) = \Omega(\int_1^n log(x) \cdot dx) = \Omega(n \cdot log(n))$.

9. Prove that, when a binary tree with n nodes is implemented using links to the left and right child, then there will be a total of $n+1$ null links.

We prove by induction. When $n=1$ there is only the root which has a null left child and null right child hence it has 2 null links, hence $P(1)$ holds. we assume it holds for all integers $i \leq k$ for some positive integer k and show it must hold for the $k+1$ integer.

Case 1: the root only has child. WLOG we assume it is the left child. the left subtree has k elements and by the inductive hypothesis has $k+1$ null links. Thus we have $(k+1)+1$. Which proves the $1st$ case.

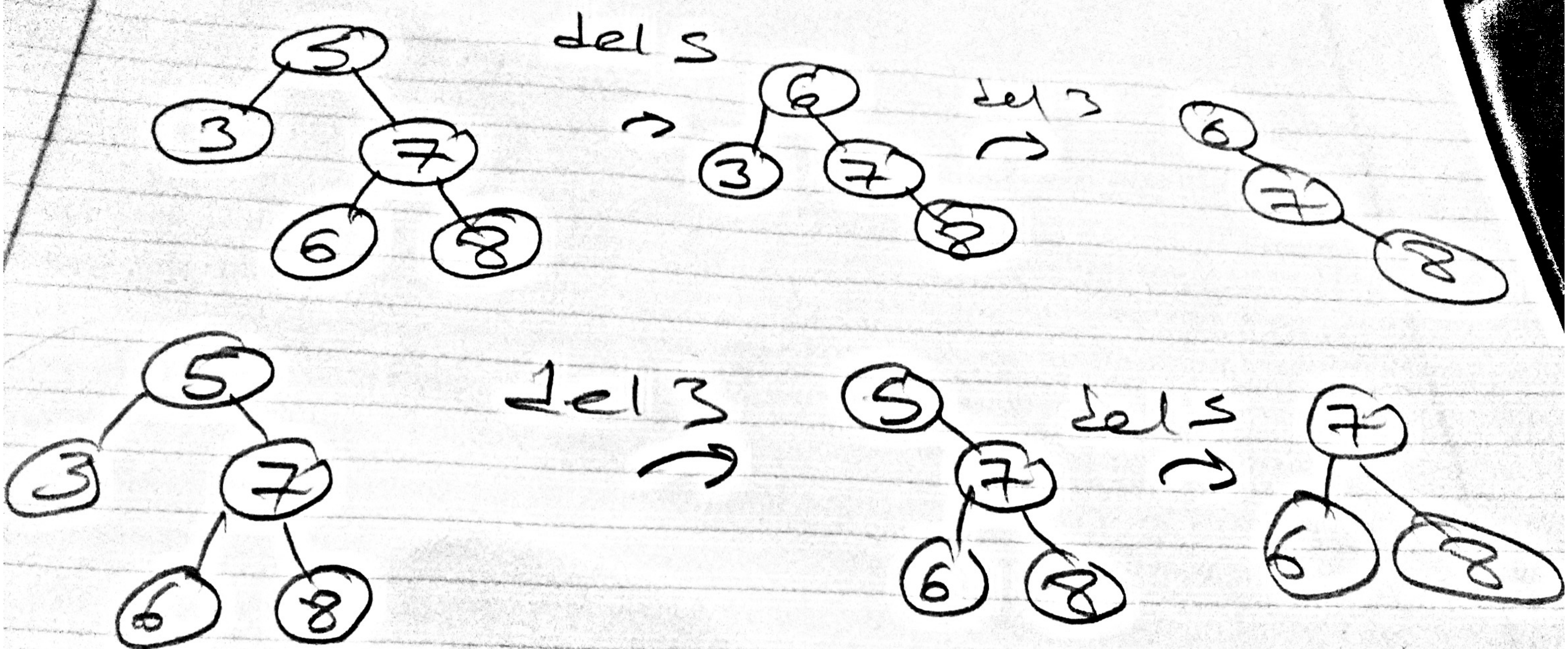
Case 2: the root has both children. Let the left subtree contain $k-i'$ elements and the right subtree contain i' elements where $0 \leq i' \leq k$. by the inductive hypothesis the left subtree contains $k-i'+1$ null links and the right subtree contains $i'+1$ null links. we have in total $(k+1)+1$ null links which proves the $2nd$ case.

Since both cases hold, $P(k+1)$ is true. We conclude by the principle of mathematical induction the result holds for all integers $n \geq 1$.

10. A full node for a binary tree is one that has two children. Prove that the number of full nodes plus one equals the number of leaves on a binary tree.

The case when there are no full nodes is when there is only the root. we have one leaf which shows $P(0)$ holds. We assume that a tree with n full nodes has $n+1$ leaves and show that $P(n+1)$ must also be true. We find one of the full nodes that contains two leaves as children. If we cannot find a node we remove a leaf from a non-full node. since the number of leaves and number of full node have not change we continosly do so until we reached the case of a full node that contains two leaves as children. We remove one of the leafs from the full node and we now have a tree with n full nodes. by the induction hypothesis we have $n+1$ leaves. When we add back the leaf we have an extra full node and an extra leave which make $P(n+1)$ hold. By the principle of mathematical induction the result holds for all integers $n \geq 0$.

11. Prove or disprove: deleting keys x and y from a BST is commutative. In otherwords, it does not matter which order the keys are deleted. the final trees will be identical. If true, provide a proof. If false, provide a counterexample.


del 5
del 3
del 3
del 5

12. What is the minimum number of nodes that a balanced tree of height 5,10,15 can have?

$\mathcal{N}(n)$	$\mathcal{N}(n-1)+\mathcal{N}(n-2)+1$
$\mathcal{N}(0)$	1
$\mathcal{N}(1)$	2
$\mathcal{N}(2)$	4
$\mathcal{N}(3)$	7
$\mathcal{N}(4)$	12
$\mathcal{N}(5)$	20
$\mathcal{N}(6)$	33
$\mathcal{N}(7)$	54
$\mathcal{N}(8)$	88
$\mathcal{N}(9)$	143
$\mathcal{N}(10)$	232
$\mathcal{N}(11)$	376
$\mathcal{N}(12)$	609
$\mathcal{N}(13)$	986
$\mathcal{N}(14)$	1596
$\mathcal{N}(15)$	2583

Hence we have that minimum nodes for a tree of height 5,10, and 15 is 20, 232 and 2583 respectively.

15. Prove that if keys $1, 2, \ldots, 2^k - 1$ are in order into an intially empty AVL tree, then the resulting tree is perfect.

We prove by induction. $P(1)$ holds trivially. Assume that the statement holds for all positive integers $i \leq k$ where k is also a positive integer. We show $P(k+1)$ is true. We insert the first $2^k - 1$ integers into the AVL tree. By the induction hypothesis this results in an perfect tree. The root is the 2^{k-1} element and having height $k-1$ with the left and right subtree having $2^{k-1} - 1$ element each and heights $k-2$. We again add 2^{k-1} next elements to the right subtree resulting in $2^k - 1$ elements and having height $k-1$. Now the left subtree and right subtree differ by 1. Adding the next element would result in a left rotation resulting in the root now being the 2^k element and the left subtree containing $2^{k-1} + 2^{k-1} - 1 = 2^k - 1$ elements and is perfect, with the right subtree now containing 2^{k-1} nodes. We now add the remaining $2^{k-1} - 1$ keys. Ignoring the root and left subtree it is identical to the case where we have inserted the integers $2^k+1, 2^k+2, \ldots, 2^k+2^{k-1}-1$. By the induction hypothesis this result in the right subtree having $2^k - 1$ elements and is also perfect. Thus we have a perfect binary tree and $P(k+1)$ holds. By the principle of mathematical induction the result holds for all integers $n \geq 1$.