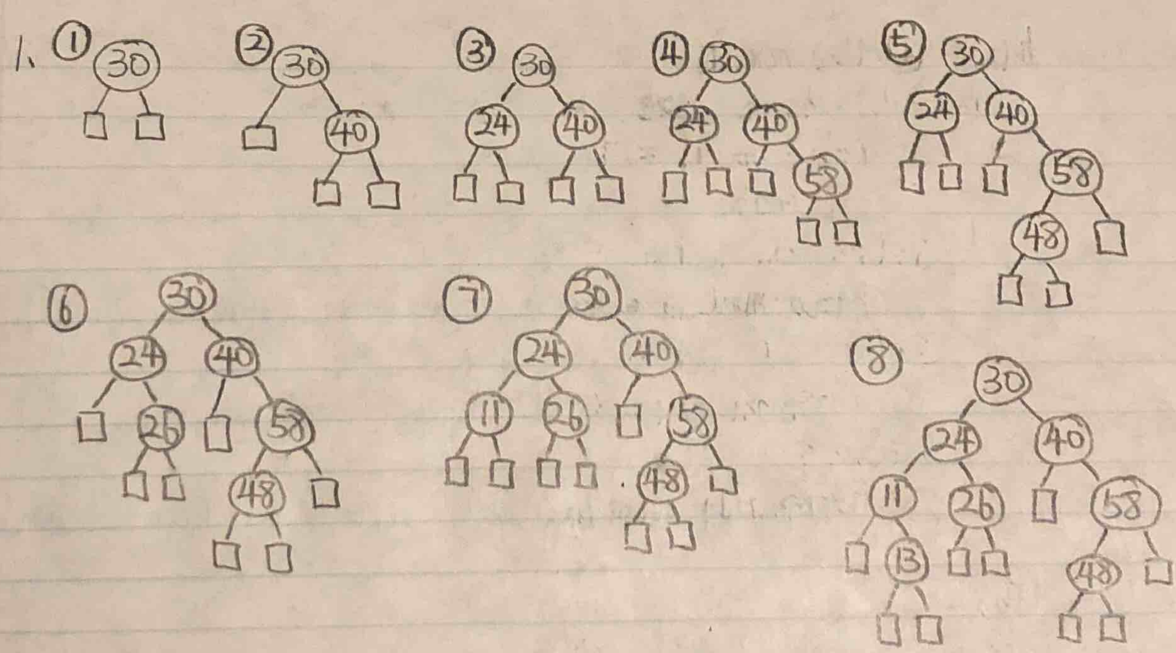
CSC 225 Assignment 3

1.



2. $\because$ For a comparison-based algorithm to build a BST on n nodes, it has permutation of n nodes of leaves.

$\therefore$ # of leaves $= \frac{n!}{(n-n)!} = n!$ (at least $n!$ leaves)

$\because$ For a tree height h has most 2^h leaves

$\therefore$ $n! \leq 2^h$

$\log_2(n!) \leq h$

$\therefore$ h is $\Omega(\log(n!))$

$\because$ h is the # of comparisons and $\log(n!)$ is the lower bound

$\therefore$ No comparison-based algorithm can build a BST on n nodes using fewer that $\log N!$ comparisons.

3.

```
countAllInRange(root, k1, k2){
    if (root <- null){
        return 0;
    }else if ( root < k1){
        return countAllInRange(root.right, k1, k2);
    }else if (root > k1) {
        return countAllInRange(root.left, k1, k2);
    }else{
        return 1+ countBiggerthan(root.left, k1) + countSmallertan(root.right, k2);
    }
}

countBiggerthan(node, k){
    if (node <- null){
        return 0;
    }else if (node < k){
        return countBiggerthan(node.right, k);
    }else{
        return 1+ countBiggerthan(node.left, k) + # of node(node.right);
    }
}

countSmallerthan (node, k){
    if (node<- null) {
        return 0;
    }else if (node > k) {
        return countSmallerthan(node.left, k);
    }else{
        return 1+ counterSmallerthan(node.right, k) + # of node(node.left);
    }
}
```

CSC 225 Assignment 3

4. $h(i) = (2i+5) \bmod 11$

Given 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5

$h(12) = (2(12)+5) \bmod 11 = 7$
$h(44) = (2(44)+5) \bmod 11 = 5$
$h(13) = (2(13)+5) \bmod 11 = 9$
$h(88) = (2(88)+5) \bmod 11 = 5$
$h(23) = (2(23)+5) \bmod 11 = 7$
$h(94) = (2(94)+5) \bmod 11 = 6$
$h(11) = (2(11)+5) \bmod 11 = 5$
$h(39) = (2(39)+5) \bmod 11 = 6$
$h(20) = (2(20)+5) \bmod 11 = 1$
$h(16) = (2(16)+5) \bmod 11 = 4$
$h(5) = (2(5)+5) \bmod 11 = 4$

Index	Value
0	11
1	39
2	20
3	5
4	16
5	44
6	88
7	12
8	23
9	13
10	94

5. $h(k,i) = (h_1(k) + i h_2(k)) \bmod t$

Let $h_1(k)=a$ and $h_2(k)=b$, the the sequence is a, $a+b$, $a+2b$ .... $a+(t-1)b$ each term mod t. The value of $h_2(k)$ must be relative prime to the hash table t for the entire hash table to be searched, otherwise let g be the gcd of t and b, $g>1$, so the search for k keys is $\frac{1}{g}$ of hash table is examined.

$\because$ t is even and $h_2(k)$ is even for some key k, the gcd for t and $h_2(k)$ is $2n$, we don't need to know what's the value of n, because their gcd at least is 2, $\therefore$ the probe sequence for k examines at most $\frac{1}{2}$ of the slots in the table before returning to slot $h_1(k)$.