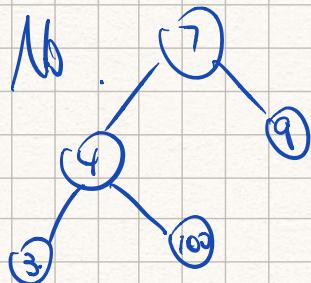
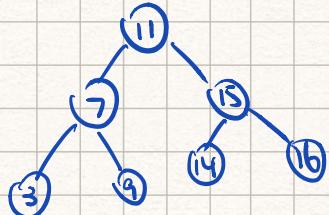


Example.

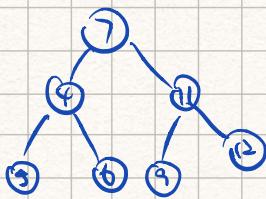
Q1.

for all nodes, the left subtree of a node

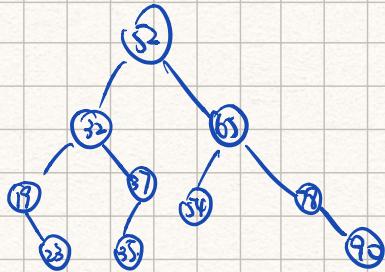
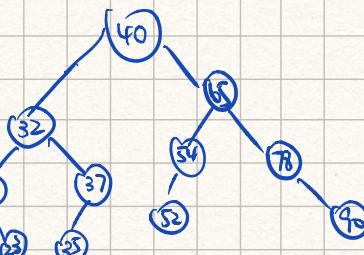
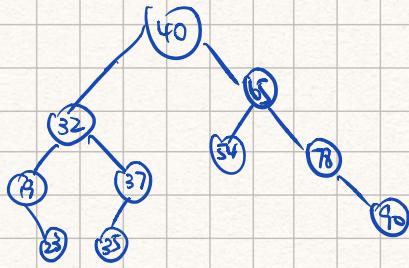
contains all values smaller
than the node
right -- bigger.



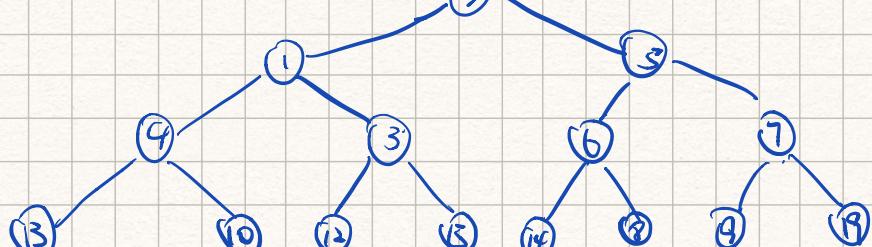
7, 4, 3, 6, 11, 9, 12

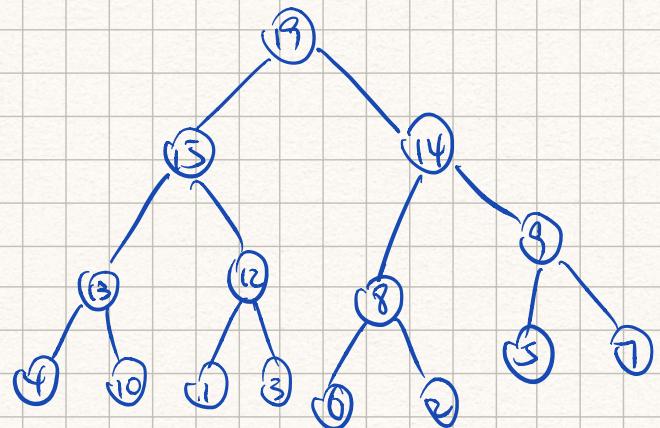
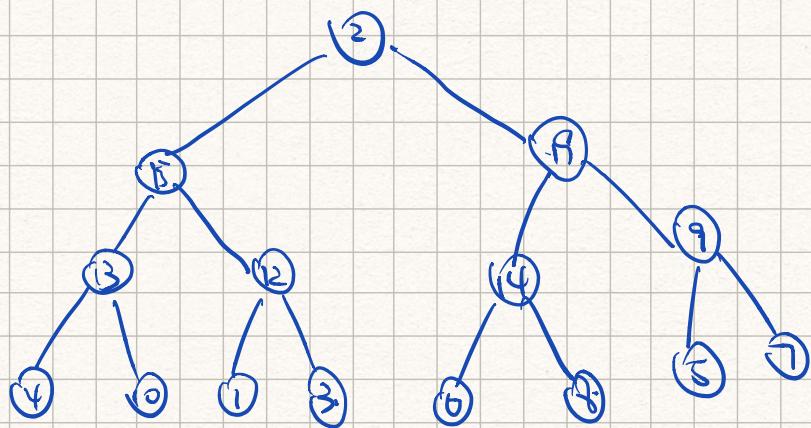
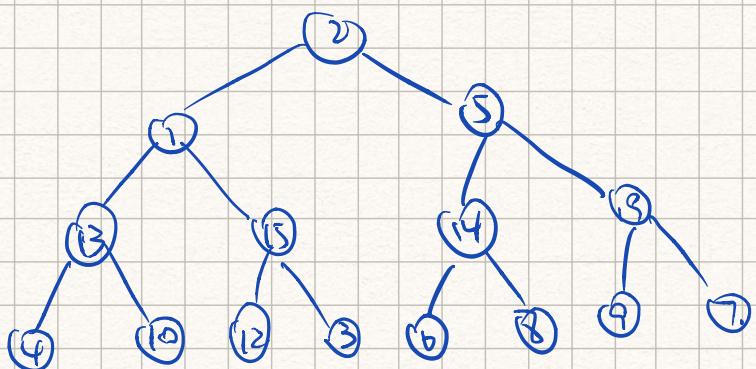
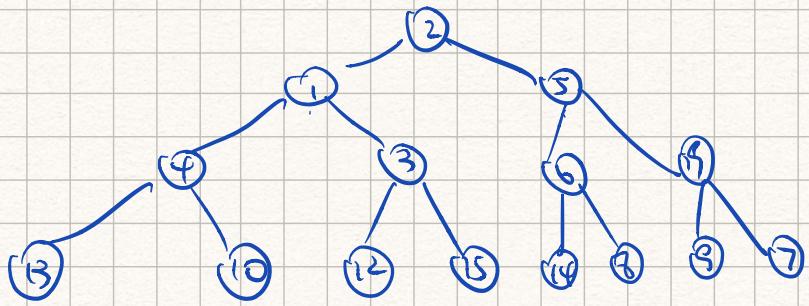


Q2.



Q3





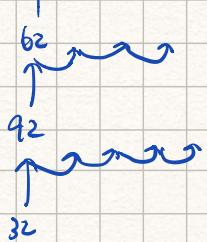
Q4.

0	1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	---	----	----	----

22 62 64 92 32 17



n $\underline{n/2}$



0	1	2	3	4	5	6	7	8	9	10	11	12
22	62	64		92	17				92.			

Arrows point from 62 to 64, 92 to 17, and 32 to 92.

0	1	2	3	4	5	6	7	8	9	10	11	12
22				64	17.							

Arrows point from 32 to 64, 32 to 17, and 32 to the end of the array.

QS

int getLength (String A)

int i=0;

while (A[i] != y)
 i++;

return i |;

bool UpperCase (String A).

return UpperHelp (String A, getLength(A)-1)

bool UpperHelp (String A, int loc).

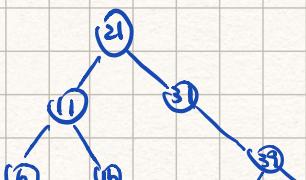
if (loc < 0). return true;

if (isUpper(A[loc])) return false

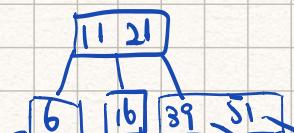
else return

UpperHelp (String A, loc-1)

Q6

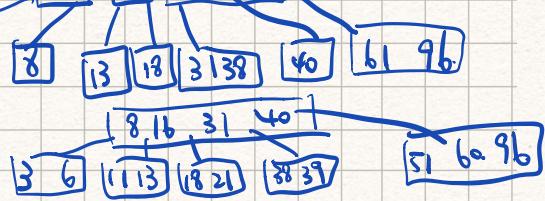


B-tree order 1. min 1 key
max 2 keys



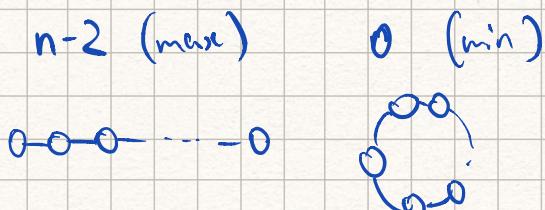


order 2.
min 2 key
max 4 key.



```
int get-requested-value (node *root, node *x)
{
    int max=-1;
    bool flag = false;
    special-in-order (root, x, &max, &flag);
    return max;
}.
```

```
void special-in-order (root, x, int* max, int* flag) {
    if (root == NULL && flag == false) {
        S-- (root->left -- );
        if (root == x)
            *flag=1;
        return;
        if (root->key < x->key)
            S-- (root->right -- );
    }
    return;
}.
```



if we DFS a tree, we can always have leaves

in DFS trees not on articulation point.

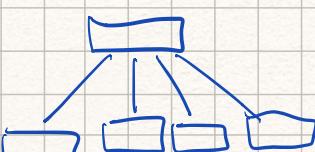
a tree with two vertices have at least two leaves.

Q7.

a. 1. $O(\log n)$

2. $O(n^{1.25})$

3. $O(n (\log n)^2)$.



d. $O(\log n)$.

b. $A \rightarrow B(n)$ $B = O(n^{\frac{1}{2}})$. B is both

$$0.01 \times 10^9 = 10^7$$

$$500\sqrt{10} = 500 \cdot 10^{\frac{1}{2}} = 500 \cdot 10^4 \cdot 10^{\frac{1}{2}} = 5 \times 10^6 \times \sqrt{10}.$$

$$5\sqrt{10} > 10$$

A is better

Q3. a. $O(n^5)$.

$$\begin{aligned} & (1+2+3+\dots+n^2-1) \times n \\ &= \underbrace{(1+(n^2-1))}_{\sum} \times \underbrace{(n^2-1)}_{\sum} = \frac{n^2(n^2-1)}{\sum} \cdot n = \frac{n^4-n^2}{\sum} \cdot n = \frac{n^5-n^3}{\sum} \end{aligned}$$

$$O(n^3),$$

$$1 + \left(\left(- \left(1 + 2 \right) \right) + \left(\left(1 + \left(1 + 2 \right) \right) + \left(1 + 2 + 3 \right) \right) \right) + \left(1 + \dots + \left(1 + \dots + n \right) \right)$$

$$\Theta(n^2).$$

$$n + n - 1 + \dots + 1 = \frac{n^2 + n}{2}.$$

Q9.

NEVER ODD OR EVEN.

E	4	1
N	2	010
Z	2	011
Z	2	0000
R	2	0001
O	2	001
D	2.	01

