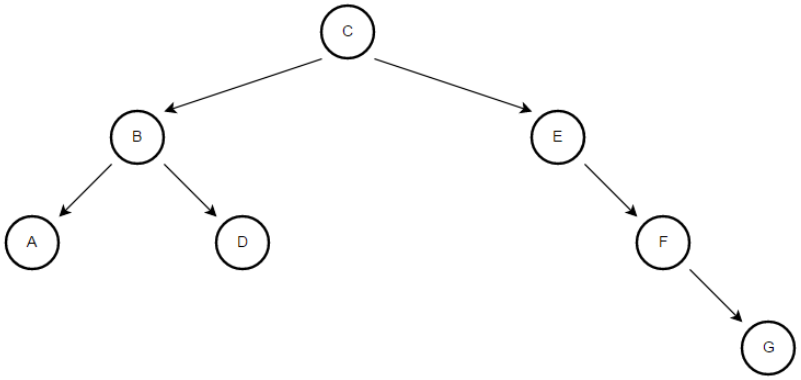
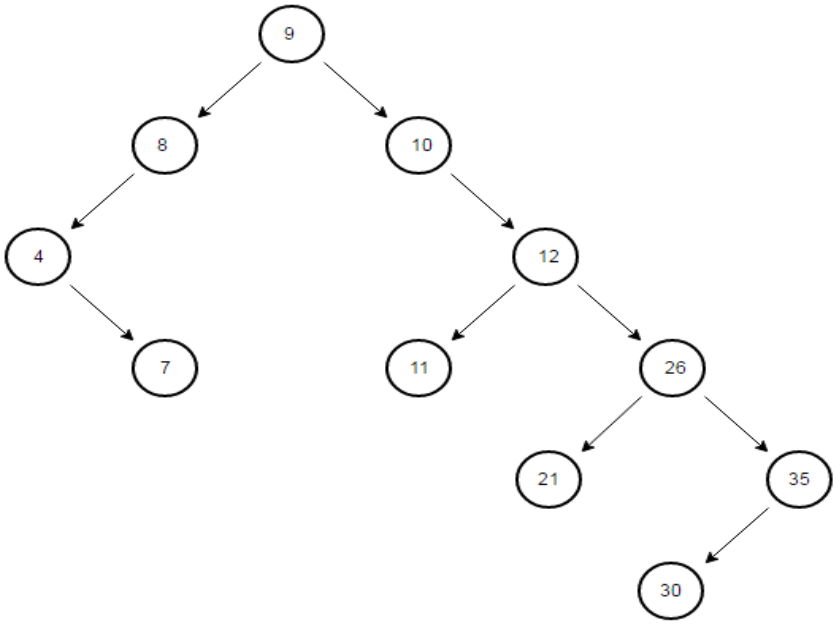
Taha Zahit Hakyemez 20160702080

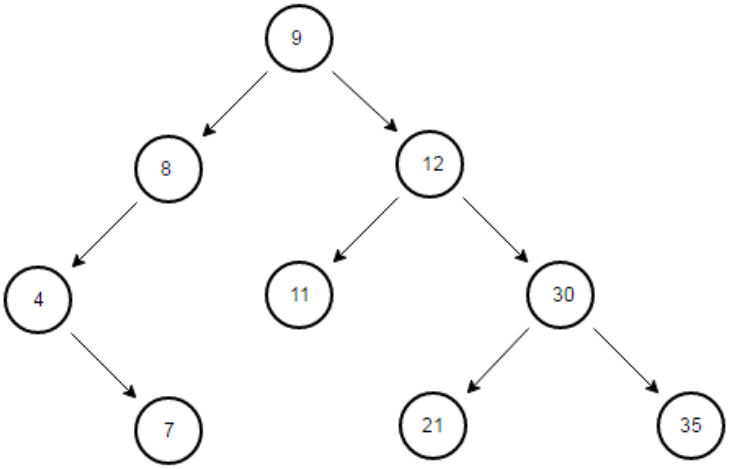
1-a)



1-b)



1-c)



2-a)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0	0	1	0	0	0	1	0	0
2	0	0	0	0	1	0	0	1	0
3	0	1	0	0	0	1	0	0	0
4	1	0	0	0	0	1	1	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0
7	0	1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	1
9	0	0	0	1	0	0	0	0	0

2-b) 1-3-7-2-6-5-8-9-4

2-c) 5-7-6-4-9-8-2-3-1

2-d)

int count = 0;

for(int i=0;i<A.size();i++) cout += A[i]!=0; //A arrayini tek boyutlu olarak aldım

return count;

2-e)$O(n^2)$

2-f)O(n)

3-a)

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	4	10	7	18	11	19	14	41	25	28	12	67	37	52	22

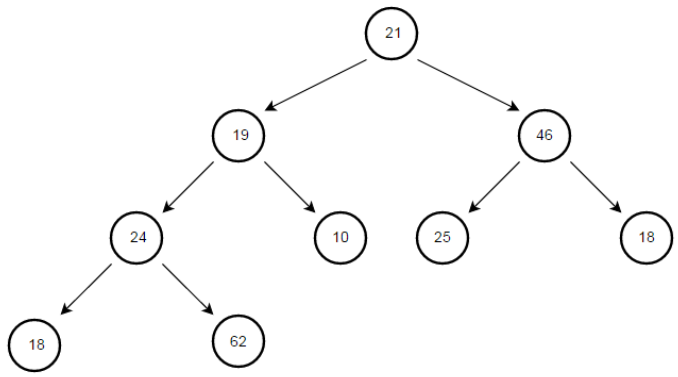
3-b)

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	7	10	14	18	11	19	22	41	25	28	12	67	37	52	

3-c)

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	7	10	14	18	11	19	19	41	25	28	12	67	37	52	22

3-d)



4-a)

```
int count(node *n)
{
        if (t == 0)   return 0;
        return 1 + count(n->left) + count(n->right);
}
```

4-b)

```
İnt func(tree & t)
{
        node * n = t.root;
        while(n->right!=null) n=n->right;
        int max= n->info;
        n=t.root;
        while(n->left!=null) n=n->left;
        int min= n->info;
        return min+max;
}
```