Question 1.

E K M P S J B F H G C Q L X U D A

Question 2.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
parent	1	/	0	1	3	1	1	1	2	3	1	0	/	1	1	/

Question 3.

1) **height[*node*] stores the height of the tree containing *node*. array[*node*] stores id of the parent of *node***

```
public void Height_union (int a, int b)
{
    Integer root1 = FIND(a);
    Integer root2 = FIND(b);
    if (root1 != root2)
    {
        if (height[root1] >= height[root2])
        {
            array[root2] = root1;
            if(height[root1] == height[root2])
            {
                height[root1]++;
            }
        }
        else
        {
            array[root1] = root2;
        }
```

```
    }
}
```

2)

When the height of two trees are the same

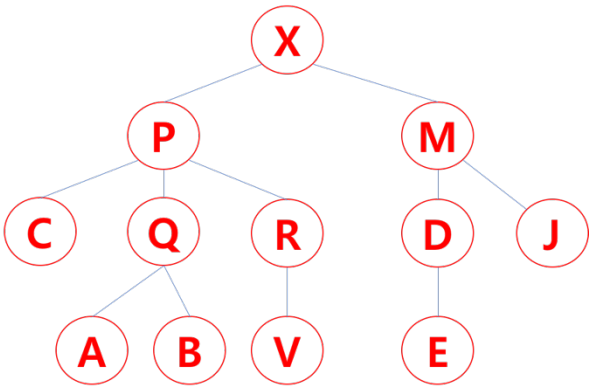
Question 4.

List of children: $\frac{3P+I}{D+3P+I}$

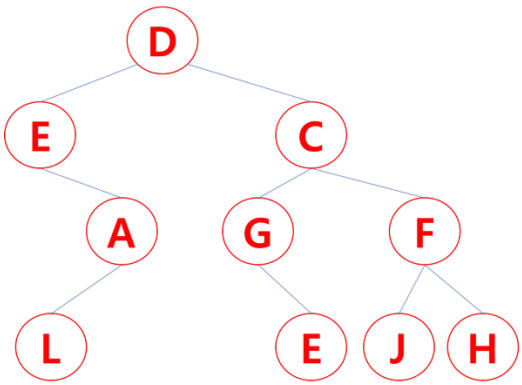
Left-child/right-sibling: $\frac{3P}{D+3P}$

Question 5.

1)



2)


D
E
C
A
G
F
L
E
J
H