

Lösningsförslag till tentamen

P r e l i m i n ä r

Kursnamn
Tentamensdatum

Algoritmer och datastrukturer
2015-06-05

Program
Läsår
Examinator

DAI2+I2
2014/2015, lp 4
Uno Holmer

Uppgift 1 (10 p)

Ingen lösning ges. Se kurslitteraturen.

Uppgift 2 (10 p)

Med ackumulerande parameter:

```
private static List<Character> getLeaves(TreeNode t, List<Character> l)
{
    if ( t == null )
        return l;
    else if ( t.isLeaf() ) {
        l.add(t.element);
        return l;
    } else {
        List<Character> leftLeaves = getLeaves(t.left, l);
        return getLeaves(t.right, leftLeaves);
    }
}
```

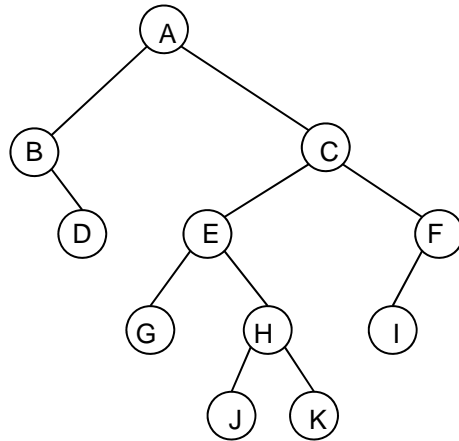
```
public static List<Character> getLeaves(TreeNode t) {
    return getLeaves(t, new ArrayList<Character>());
}
```

eller

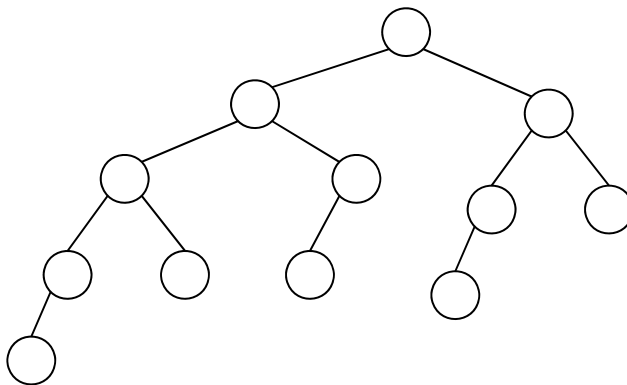
```
private static List<Character> getLeaves(TreeNode t) {
    if ( t == null )
        return new ArrayList<Character>();
    else if ( t.isLeaf() ) {
        List<Character> l = new ArrayList<Character>();
        l.add(t.element);
        return l;
    } else {
        List<Character> leftLeaves = getLeaves(t.left);
        List<Character> rightLeaves = getLeaves(t.right);
        leftLeaves.addAll(rightLeaves);
        return leftLeaves;
    }
}
```

Uppgift 3 (2+1 p)

a) Trädet måste med de givna genomlöpningsordningarna ha formen



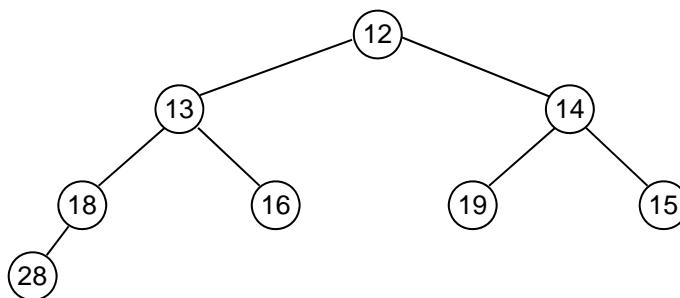
b) Ett AVL-träd med minimalt antal noder av djup 4:



Uppgift 4 (2+3 p)

a) Trädet skall vara komplett och uppfylla ordningsvillkoret. Se kursboken.

b)



Uppgift 5 (6+6 p)

- a) Noderna besöks i denna ordning, approximerade avstånd inom parenteser:
B (0), F (5), G (14,12), D (16,15,14), A (40), C (42,41), E (60,59,58) .
- b) ABCDEF, ABCDFE, ABDCEF, ABDCFE, ABDFCE, BACDEF, BACDFE, BADCEF, BADCFE, BADFCE, ACBDEF, ACBDFE.

Uppgift 6 (4+4 p)

a)

De fyra första talen sätts in på platserna:

0	
1	32
2	
3	
4	18
5	40
6	25

b)

Tabellen är nu mer än halvfull. Då man försöker sätta in 4 terminerar inte sökningen efter en ledig plats. Vi hashar om till en tabell med storlek minsta primtal större än $2 \cdot 7$ vilket är 17.

0	
1	18
2	
3	
4	4
5	
6	40
7	
8	25
9	
10	
11	
12	
13	
14	
15	32
16	

Uppgift 7 (12 p)

```
public static List<Edge> minimumSpanningTree(List<Edge> edges) {
    Queue<Edge> pq = new PriorityQueue<>(new EdgeComparator());
    for ( Edge e : edges )
        pq.add(e);
    int n = noOfNodes(edges);
    DisjointSets ds = new DisjointSets(n);
    int noOfUnions = 0;
    List<Edge> result = new ArrayList<>();
    while ( noOfUnions < n-1 ) {
        Edge e = pq.poll();
        int s1 = ds.find(e.src);
        int s2 = ds.find(e.dest);
        if ( s1 != s2 ) {
            ds.union(s1,s2);
            result.add(e);
            noOfUnions++;
        }
    }
    return result;
}

private static int noOfNodes(List<Edge> edges) {
    Set<Integer> nodes = new HashSet<>();
    for ( Edge e : edges ) {
        nodes.add(e.src);
        nodes.add(e.dest);
    }
    return nodes.size();
}
```