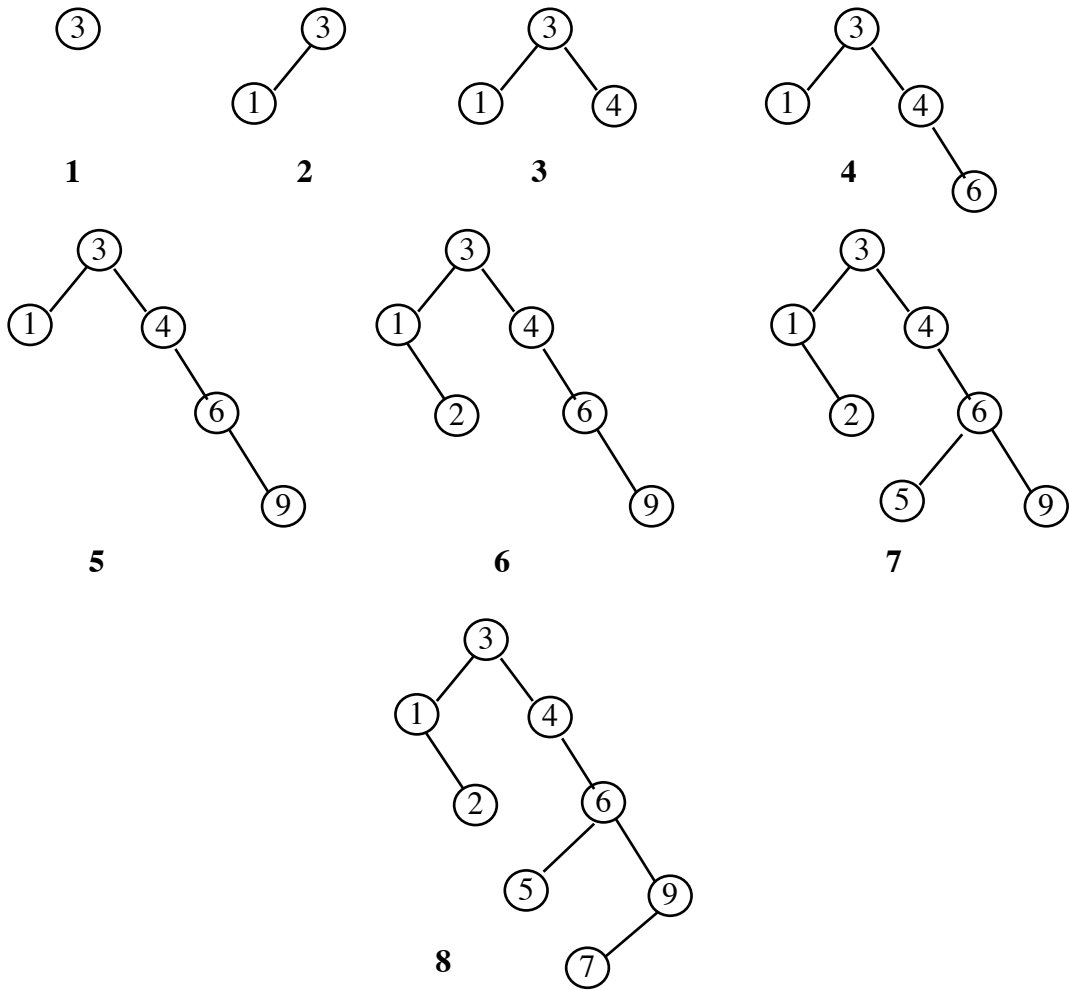


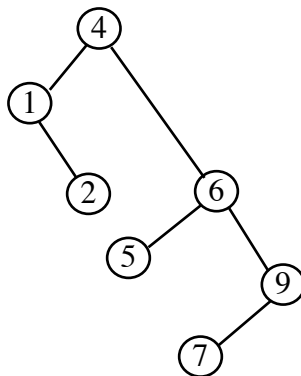
Opgaveløsninger (sæt 11)

Opgave 41: 19.1 (1 point)

(a)

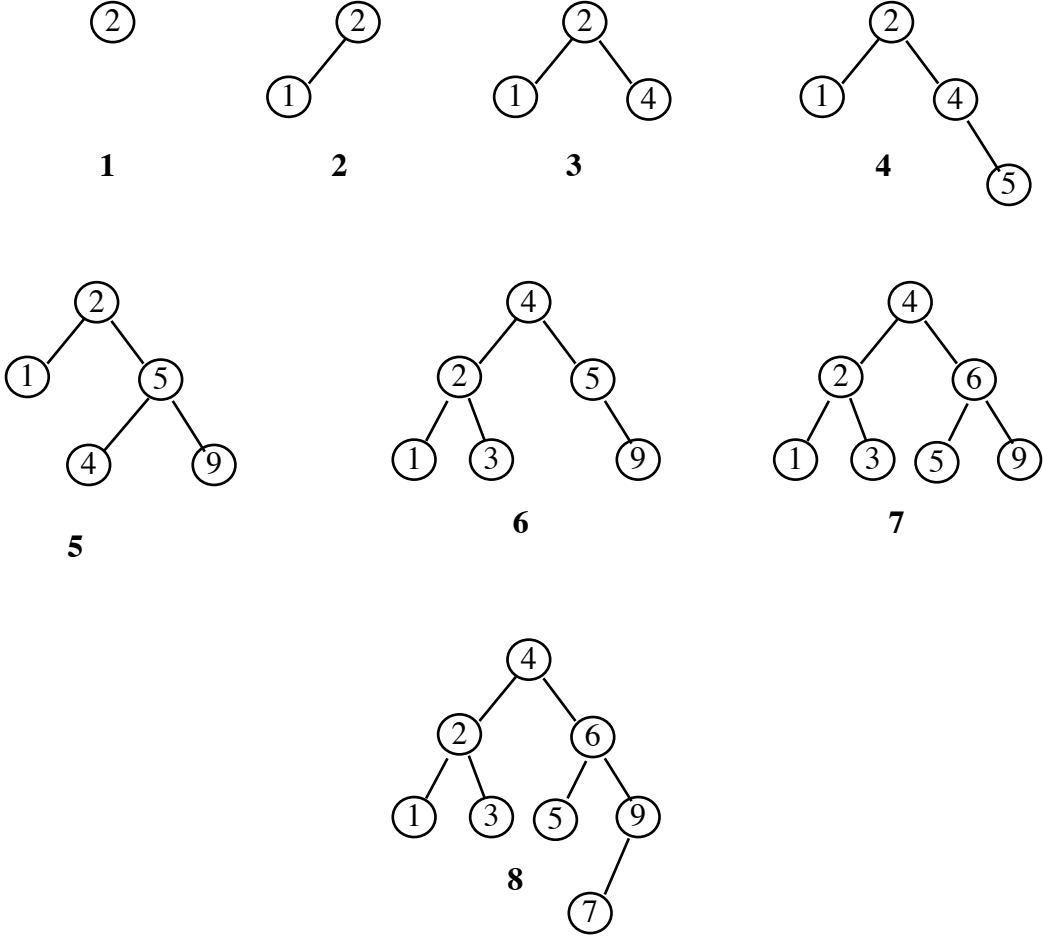


(b) sletning af roden (3 erstattes med den mindste knude i rodens højre undertræ: 4)

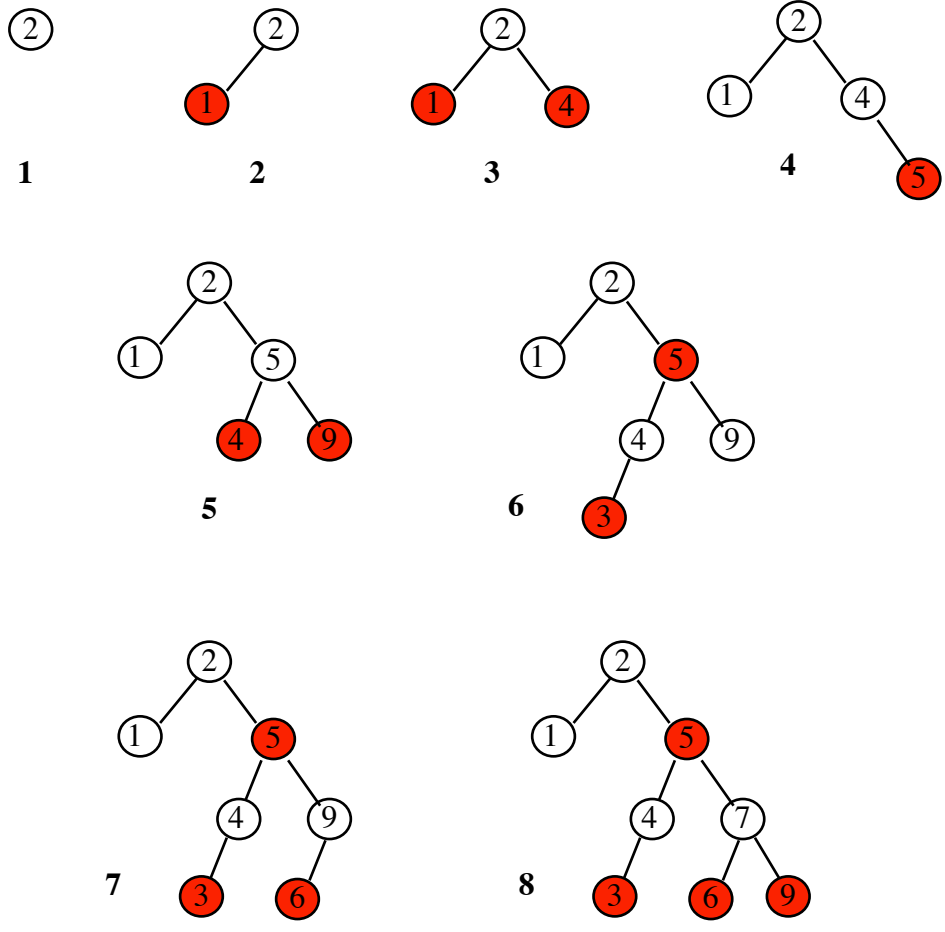


Opgave 42: 19.5 (3 point, ikke-obligatorisk)

(a) AVL-træ



(b) Rød-sort-træ

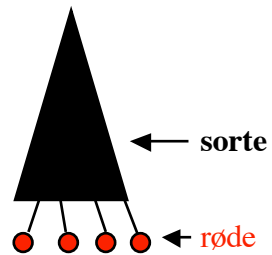


Opgave 43: 19.11 (2 point, ikke-obligatorisk)

(a)

Alle blade i et AVL-træ må nødvendigvis være på et eller to niveauer.

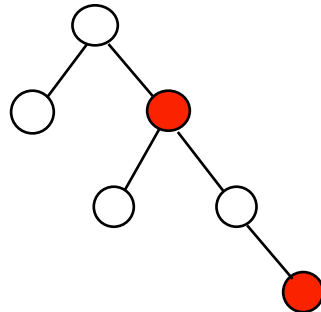
Farv alle knuder **sorte**, med undtagelse af bladene på det nederste niveau, som farves **røde**.



Det ses let, at dette træ er et rød-sort-træ.

(b)

Nedenstående rød-sort-træ er ikke et AVL-træ (højdebalancen er 2 i roden).



Opgave 44: 19.15 (2 point)

(a)

```
private BinaryNode find(Comparable x, BinaryNode t) {
    if (t != null) {
        if (x.compareTo(t.element) < 0)
            return find(x, t.left);
        if (x.compareTo(t.element) > 0)
            return find(x, t.right);
    }
    return t;
}
```

(b)

```
private BinaryNode findMin(BinaryNode t) {
    return t != null && t.left != null ? findMin(t.left) : t;
}
```

(c)

```
private BinaryNode findMax(BinaryNode t) {
    return t != null && t.right != null ? findMax(t.right) : t;
}
```

Opgave 45: 19.17 (1 point)

```
private BinaryNode findKth(int k) {
    BinaryNode t = root;
    while (t != null) {
        int leftSize = t.left != null ? t.left.size : 0;
        if (k <= leftSize)
            t = t.left;
        else if (k == leftSize + 1)
            return t;
        else {
            k -= leftSize + 1;
            t = t.right;
        }
    }
    throw new IllegalArgumentException();
}
```