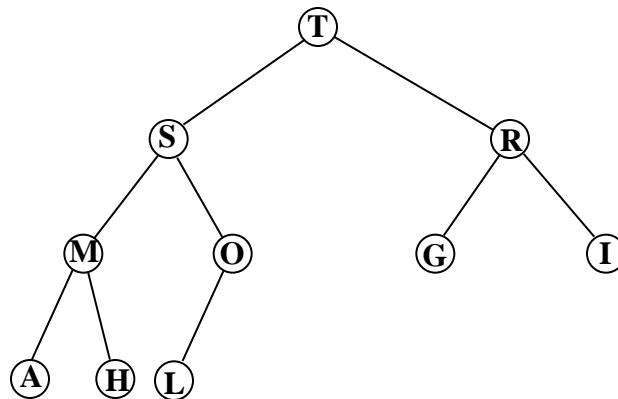


Vejledende løsninger til opgave 1, 2 og 3

Opgave 1: Træer

Spørgsmål 1.1



Spørgsmål 1.2

```
root = new Node('T',  
                new Node('S',  
                        new Node('M',  
                                new Node('A', null, null),  
                                new Node('H', null, null)  
                        ),  
                        new Node('O',  
                                new Node('L', null, null),  
                                null  
                        )  
                ),  
                new Node('R',  
                        new Node('G', null, null),  
                        new Node('I', null, null)  
                )  
);
```

Spørgsmål 1.3

```
Node makeTree(char a[], int N) {
    return makeSubtree(a, 1, N);
}

Node makeSubtree(char a[], int i, int N) {
    return i > N ? null :
           new Node(a[i], makeSubtree(a, 2*i, N),
                   makeSubtree(a, 2*i+1, N));
}
```

Opgave 2: Stakke og køer

Spørgsmål 2.1

Der benyttes to hjælpestakke, x1 og x2.

```
Stack x1 = new Stack();
while (!s.isEmpty())
    x1.push(s.pop());
Stack x2 = new Stack();
while (!x1.isEmpty())
    x2.push(x1.pop());
while (!x2.isEmpty())
    s.push(x2.pop());
```

Spørgsmål 2.2

Der benyttes en hjælpestak, x.

```
Stack x = new Stack();
while (!s.isEmpty())
    x.push(s.pop());
Stack t = new Stack();
while (!x.isEmpty()) {
    Object item = x.pop();
    t.push(item);
    s.push(item);
}
```

Spørgsmål 2.3

Der benyttes en hjælpekeø x .

```
Queue x = new Queue();
while (!q.isEmpty())
    x.insert(q.remove());
Queue r = new Queue();
while (!x.isEmpty()) {
    Object item = x.remove();
    r.insert(item);
    q.insert(item);
}
```

Opgave 3: Rekursion

Svar: 32

Begrundelse:

$$\begin{aligned} f(1, 7) &= f(f(3, 6), f(2, 5)) = \\ &= f(f(f(5, 5), f(4, 4)), f(f(4, 4), f(3, 3))) = \\ &= f(f(10, 8), f(8, 6)) = \\ &= f(18, 14) = \\ &= 32 \end{aligned}$$

Alternativ begrundelse:

Værdien bestemmes ud fra kaldtræet:

