

Lösningsförslag till tentamen

P r e l i m i n ä r

Kursnamn
Tentamensdatum

Algoritmer och datastrukturer
2018-06-01

Program
Läsår
Examinator

DAI2+I2
2017/2018, lp 4
Uno Holmer

Uppgift 1 (10 p) Ingen lösning ges. Se kurslitteraturen.

Uppgift 2 (2+6 eller 8 p)

a)

```
public String toString() {  
    return "(" + leftOperand.toString() + " + "  
        + rightOperand.toString() + ")";  
}
```

(analogt för den andra klassen)

b)

I Addition:

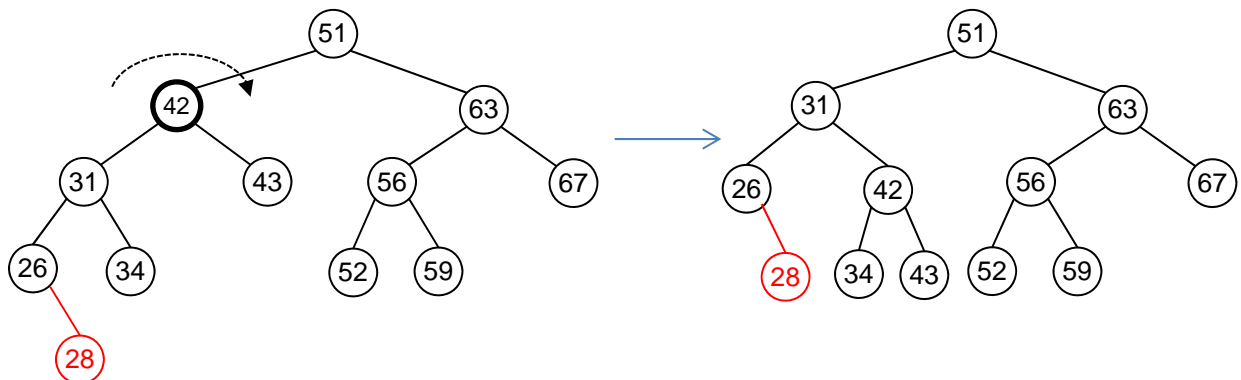
```
public Expression simplify() {  
    Expression leftSimp = leftOperand.simplify();  
    Expression rightSimp = rightOperand.simplify();  
    if ( leftSimp.isZero() )  
        return rightSimp;  
    else if ( rightSimp.isZero() )  
        return leftSimp;  
    else  
        return new Addition(leftSimp,rightSimp);  
}
```

I Multiplication:

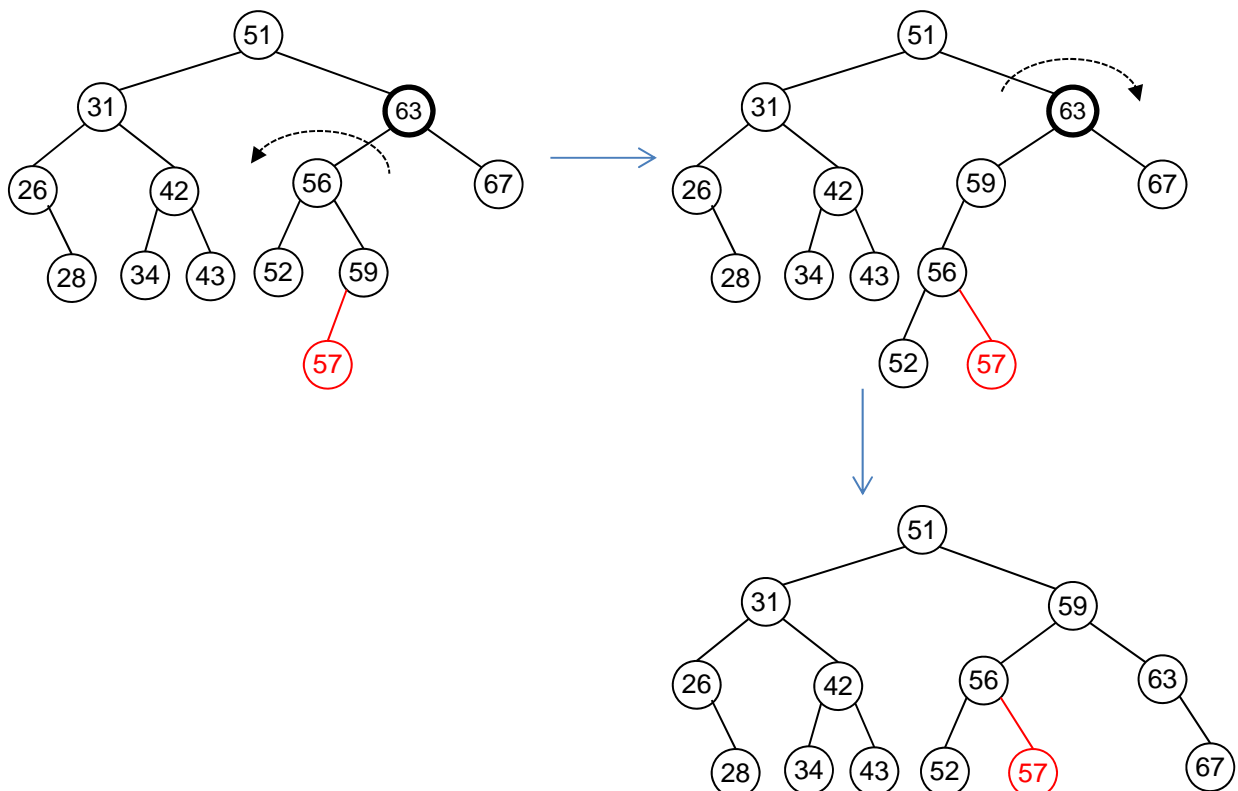
```
public Expression simplify() {  
    Expression leftSimp = leftOperand.simplify();  
    Expression rightSimp = rightOperand.simplify();  
    if ( leftSimp.isZero() || rightSimp.isZero() )  
        return new Constant("0");  
    else if ( leftSimp.isOne() )  
        return rightSimp;  
    else if ( rightSimp.isOne() )  
        return leftSimp;  
    else  
        return new Multiplication(leftSimp,rightSimp);  
}
```

Uppgift 3 (2+1+4 p)

- a) Insättninsordningarna (4, 2, 1, 3, 6, 5, 7), (4, 2, 3, 1, 6, 5, 7), (4, 2, 1, 3, 6, 7, 5), (4, 6, 5, 7, 2, 3, 1), (4, 6, 2, 7, 1, 5, 3) samt några till ger perfekt balanserade sökträd.
- b) Ett fullt träd av höjd H har $2^{H+1} - 1$ noder.
- c) Efter insättning av 28 (fall 1 i Weiss) krävs en enkel högerrotation, vilket ger trädet till höger. Den djupast liggande obalanserade noden är 42.



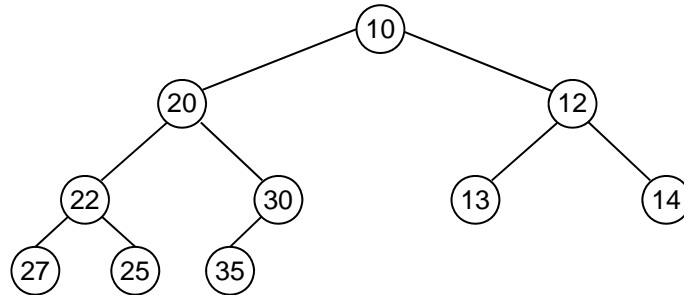
Därefter sätter vi in 57 och måste göra en dubbelrotation (Weiss fall 3).



Uppgift 4 (3+4 p)

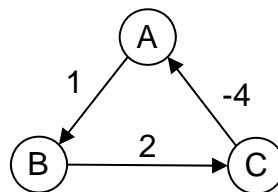
- a) Endast det vänstra trädet är en binär hög. Det mittersta är ett komplett träd men bryter mot ordningsvillkoret. Det högra är inte komplett.
b) Efter ett anrop av `buildHeap`:

$-\infty$	10	20	12	22	30	13	14	27	25	35
0	1	2	3	4	5	6	7	8	9	10



Uppgift 5 (3+6 p)

- a) I en graf med en negativ kostnadscykel kan inte Dijkstras algoritm minimera avstånden, t.ex.:



- b) Grafen skall vara en DAG, alltså en riktad graf utan cykler vilket uppfylls av den mittersta. Alla topologiska ordningar av noderna i denna är: A B C D E F, A B C E D F, A C B E D F, A C B D E F, A C E B D F, B A C D E F, B A C E D F.

Uppgift 6 (2+3 p)

- a) Tabellen skall ha primtalsstorlek och belastningsfaktorn λ skall vid insättning av ett nytt element vara högst 0.5.
b) Objektet X sätts in i position 11.

Uppgift 7 (12 p)

Vi gör en BFS-sökning med hjälp av en FIFO-kö:

```
public void levelOrder(String startVertex) {
    Queue<Vertex> q = new LinkedList<>();
    q.add(graphTable.get(startVertex));
    int dist = 0;
    System.out.print(dist + ": ");
    while ( ! q.isEmpty() ) {
        Vertex v = q.poll();
        if ( ! v.visited ) {
            if ( v.dist == dist + 1 ) { // Next level
                dist++;
                System.out.print("\n" + dist + ": ");
            }
            System.out.print(v + " ");
            v.visited = true;
            for ( Vertex w : v.neighbours ) {
                if ( w.dist == 0 )
                    w.dist = v.dist + 1;
                q.add(w);
            }
        }
    }
}
```