CHALMERS LINDHOLMEN
Institutionen för data- och elektroteknik
Uno Holmer
uno.holmer@chl.chalmers.se

Lösningsförslag till tentamen *

Kursnamn	**Algoritmer och datastrukturer, 4p**
Tentamensdatum	**2000-04-25**
Program	**DAI 2**
Läsår	**99/00, lp I/II**
Examinator	**Uno Holmer**

*** se även `http://www.chl.chalmers.se/~holmer/` där finns det mesta av kursmaterialet.**

Uppgift 1 (10 p)

Ingen lösning ges

Uppgift 2 (12 p)

a) (6 p)

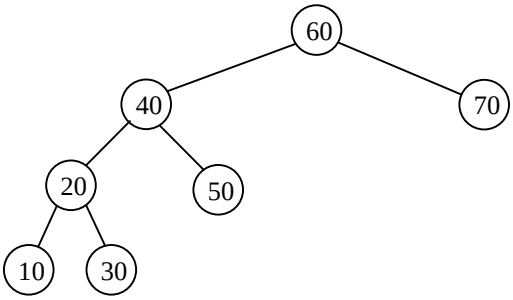
```
void Mobile::PrettyPrint() const {
    if ( Simple() )
        cout << '(' << theMass << ')';
    else {
        cout << '[';
        Left->PrettyPrint();
        cout << ',' << LeftLength << ',' << RightLength << ',';
        Right->PrettyPrint();
        cout << ']';
    }
}
```

b) (6 p)

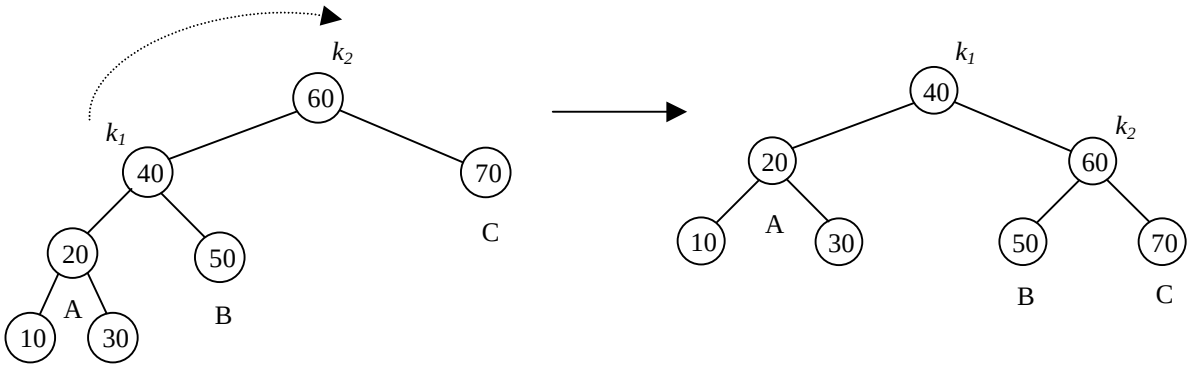
```
float
Mobile::Mass() const {
    if ( Simple() )
        return theMass;
    else
        return Left->Mass() + Right->Mass();
}

bool
Mobile::Balanced() const {
    const float eps = 0.000001;
    return Simple() ||
           Left->Balanced() && Right->Balanced() &&
           fabs( LeftLength * Left->Mass() -
                 RightLength * Right->Mass() ) < eps;
}
```

Uppgift 3 (6 p)

a) (2 p)
Om 20 sätts in först, följt av 10 och 30 i godtycklig ordning, så fås det mest höjdbalanserade trädet



b) (4 p)
Trädet i a) ovan motsvarar fall 1 i Weiss fig. 18.23 och transformeras till ett AVL-träd med en enkelrotation:



Uppgift 4 (9 p)

a) (3 p)

	A	B	C	D	E	F	G	H
A	0	∞	2	19	∞	∞	∞	∞
B	1	0	4	∞	∞	∞	∞	∞
C	∞	∞	0	15	22	3	∞	∞
D	∞	∞	∞	0	5	∞	∞	∞
E	∞	∞	∞	∞	0	∞	∞	∞
F	∞	∞	∞	∞	18	0	5	4
G	∞	∞	∞	6	∞	∞	0	∞
H	∞	∞	∞	∞	13	∞	2	0

b) (3 p)

Kortaste vägarna från B till övriga noder, samt kostnaden för resp väg inom parentes är

	A	**C**	**D**	**E**	**F**	**G**	**H**
*** oviktad**	BA (1)	BC (1)	BCD (2)	BCE (2)	BCF (2)	BCFG (3)	BCFH (3)
viktad	BA (1)	BAC (3)	BACFGD (17)	BACFGDE (22)	BACF (6)	BACFG (11)	BACFH (10)

*) I vissa fall finns alternativa vägar av samma längd.

c) (3 p)

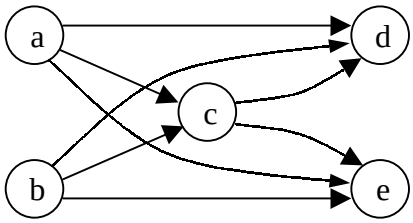
Allmänt gäller:

- om det finns en väg från x till y i grafen så kommer x före y i den topologiska ordningen
- om x kommer före y i den topologiska ordningen så kan det finnas en väg från x till y i grafen
- om x kommer före y i den topologiska ordningen så kan det inte finnas en väg från y till x i grafen

Om noderna i en graf kan ordnas topologiskt på de två olika sätten abced och bacde kan man dra följande slutsatser:

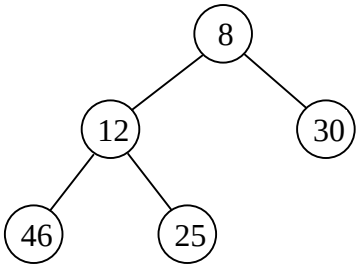
- eftersom a kommer före b i den ena ordningen, men tvärtom i den andra så kan inte a och b vara förbundna med varandra (de finns inga cykler)
- eftersom d kommer före e i den ena, men tvärtom i den andra så kan inte d och e vara förbundna med varandra

En graf som uppfyller detta fås genom att stryka valfritt antal bågar utan att grafen delas i två osammanhängande delar:

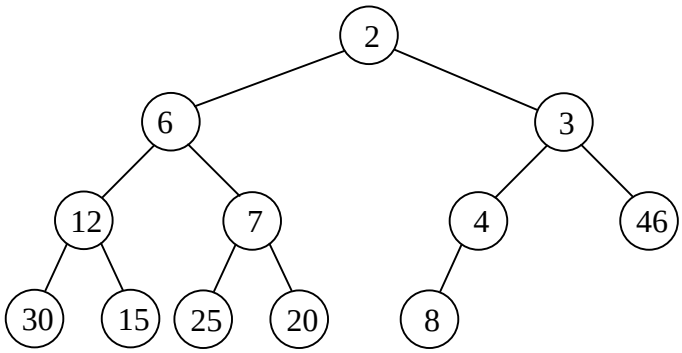


Uppgift 5 (6 p)

a) (3 p) *H* efter tre anrop av DeleteMin:



b) (3 p) *H* efter insättning av talen 15, 20, 6, 2 med Insert:



Uppgift 6 (10 p)

```
void Simulator::Run() {
    Resultat R;
    BinaryHeap<Resultat> ResQ( R );
    while ( true ) {
        Startlista >> R;
        if ( Startlista.eof() )
            break;

        if ( R.Tid == 0 )
            // Första loppet
            R.Tid = 3600 + random( 2*3600 );
        else
            // Ny tid = gammal +/- 15min
            R.Tid = R.Tid + 15*60( random( 3 ) – 1 );
        ResQ.Toss( R );
    }
    int NyttStartnr = 1;
    ResQ.FixHeap();
    while ( ! ResQ.IsEmpty() ) {
        Resultat R;
        ResQ.DeleteMin( R );
        R.Delt.Startnr = NyttStartnr++;
        NyStartlista << R;
    }
}
```

Uppgift 7 (6 p)

a) (4 p)

Sekvensen Insert(95); Remove(165); Insert(7); Insert(634); Insert(15); Insert(337); Insert(49); Insert(79); ger följande resultat. Borttagning är ”lat” och frigör ej cellen för nya element.

Linjär sondering

0	**337**
1	**49**
2	**79**
3	
4	**634**
5	*165*
6	*26*
7	**95**
8	**7**
9	**15**

= ”lazy deletion”

Kvadratisk sondering

0	**49**	$49 + 1^2 \mod 10 = 0$
1	**15**	$5 + 4^2 \mod 10 = 1$
2		
3	**79**	$79 + 2^2 \mod 10 = 3$
4	**634**	$634 \mod 10 = 4$
5	*165*	
6	*26*	
7	**7**	$7 \mod 10 = 7$
8	**337**	$337 + 1^2 \mod 10 = 8$
9	**95**	$5 + 2^2 \mod 10 = 9$

b) (3 p)

36124 hashar in i position 4 och alternativen 0, 3, 4, 5, 8, 9 är upptagna. 0, 3, 4, 5, 8, 9 är de enda alternativa positioner man kan få, d.v.s. $i^2 \equiv x \pmod 9$ för $x \in \{0,3,4,5,8,9\}$. Elementet kan alltså ej sättas in trots att det finns ledig plats. För att undvika denna situation måste $\lambda \leq 0.5$ och tabellstorleken vara ett primtal. Inget av dessa krav är uppfyllt.