

Skriftlig Eksamen
Algoritmer og Datastrukturer (DM02)

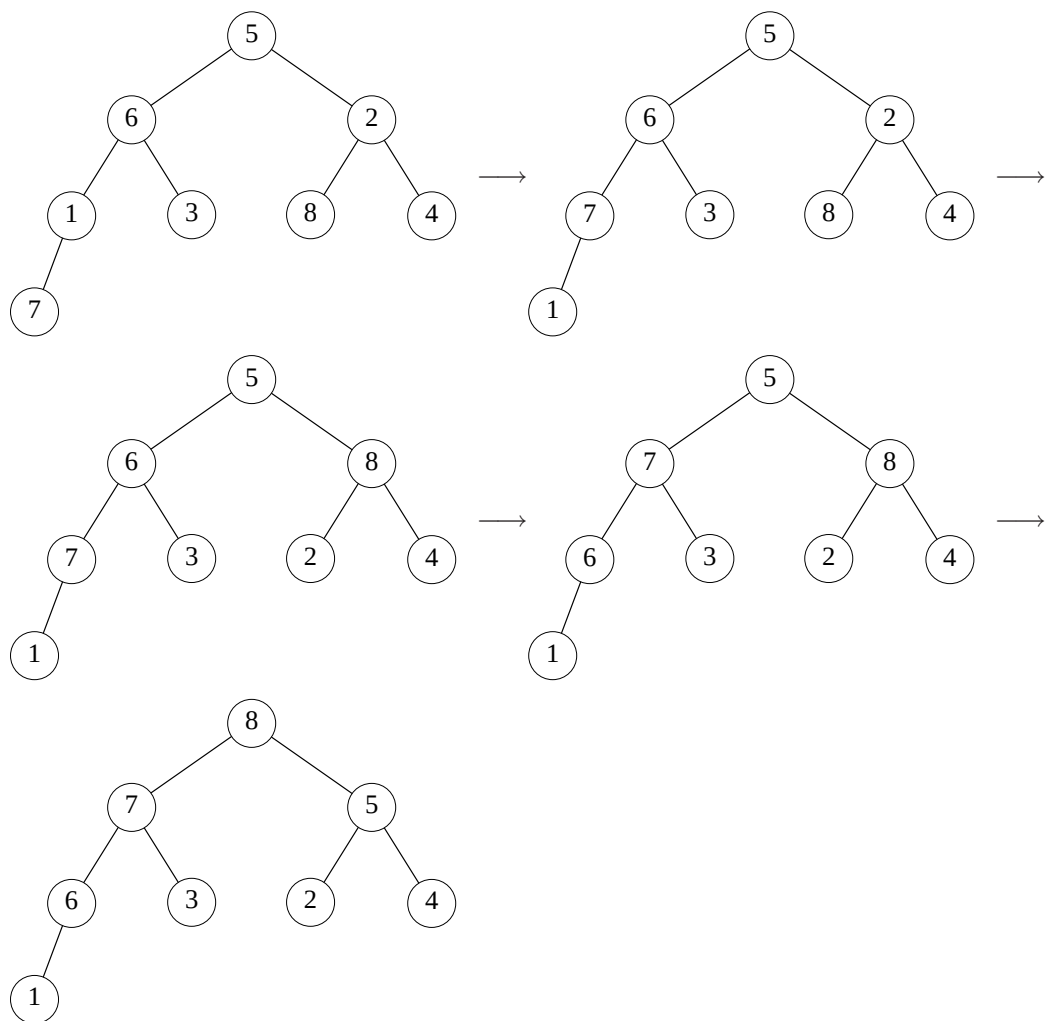
Institut for Matematik og Datalogi
Syddansk Universitet, Odense

Tirsdag den 8. januar 2008, kl. 9–13

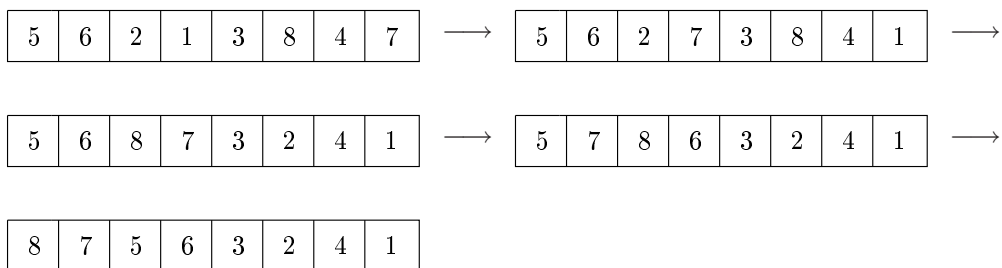
Løsningsforslag

Opgave 1

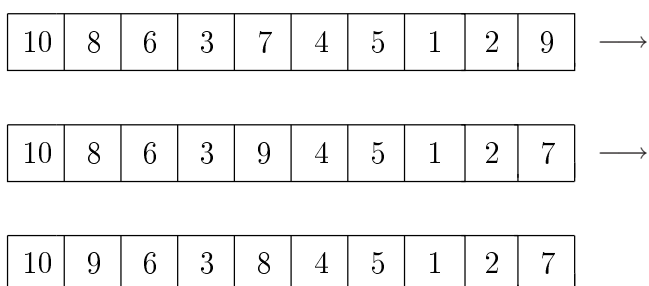
Spørgsmål a:



ELLER



Spørgsmål b:

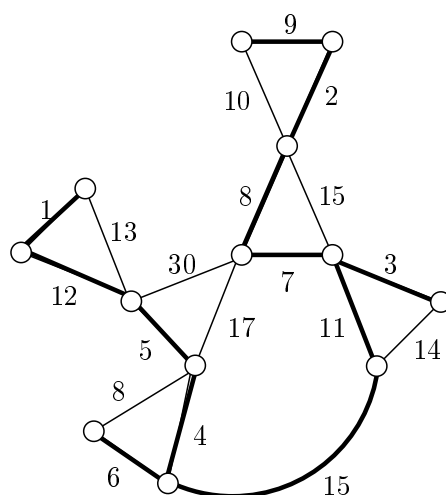


Spørgsmål c: Elementet havner på plads 6. Først prøves plads 4, derefter plads 5 og til sidst plads 6.

Opgave 2

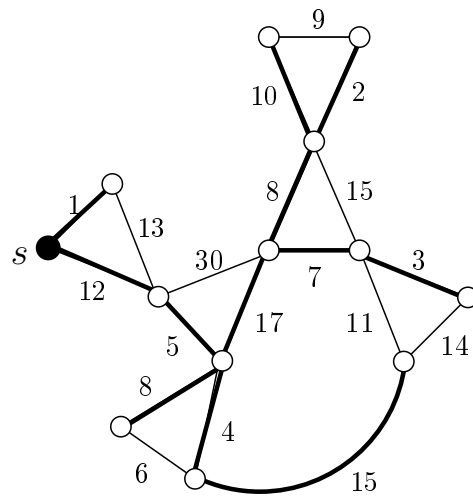
Spørgsmål a: Antag til modstrid, at der findes et MST T , som indeholder e . Hvis man fjerner e , falder T i to dele (sammenhængskomponenter). Da e tilhører en kreds C , må der være en anden kant $e' \in C$, som forbinder disse to dele. Bemærk, at e' ikke kan tilhøre T , da det ville give en kreds i T . Dermed er $T' = T + \{e'\} - \{e\}$ også et udspændende træ. Da $w(e') < w(e)$, er T' lettere end T , hvilket er i modstrid med, at T er et MST.

Spørgsmål b: Kanterne i MST er tegnet med fed:



Ifølge spm. b skal den tungeste kant i en kreds *ikke* med. Hvis vi sletter alle kanter, som er den tungeste i en kreds, indeholder grafen ikke længere nogen kreds. Dermed har vi opnået et udspændende træ.

Spørgsmål c: Kanterne i korteste-vej-træet er tegnet med fed:



Man går den korteste vej gennem hver trekant. For knuder i kredsen bestående af fire trekanter og den “ekstra kant” skal man desuden vælge den korteste vej rundt.

Opgave 3

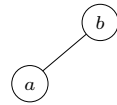
Spørgsmål a: (5,17)

Spørgsmål b: Indsættelse og sletning foretages som normalt i rød-sortede træer, med følgende tilføjelser.

Før højre-rotation om kanten (a, b) , hvor b er forælder til a :

$$a.y_{\max} \leftarrow b.y_{\max}$$

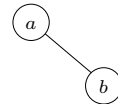
$$b.y_{\max} \leftarrow \max\{b.y, \text{right}[a].y_{\max}, \text{right}[b].y_{\max}\}$$



Før venstre-rotation om kanten (a, b) , hvor a er forælder til b :

$$b.y_{\max} \leftarrow a.y_{\max}$$

$$a.y_{\max} \leftarrow \max\{a.y, \text{left}[a].y_{\max}, \text{left}[b].y_{\max}\}$$



Efter indsættelse af en knude v udføres **I0pdater**(v):

I0pdater(v)

$p \leftarrow \text{parent}[v]$

If $p \neq \text{NIL}$

 If $v.y_{\max} > p.y_{\max}$

$p.y_{\max} \leftarrow v.y_{\max}$

I0pdater(p)

Efter sletning af en knude v :

Lad u være den strukturelt slettede knude, og lad p være u 's forælder. Efter u er fjernet, udføres **S0pdater**(p). Bemærk: da den strukturelt slettede knude er efterkommer af den logisk slettede knude, behøver vi kun bekymre os om strukturelt slettet knude.

S0pdater(v)

If $v \neq \text{NIL}$

 If $v.y < v.y_{\max}$

$y \leftarrow \text{maxy}(\text{left}[v], \text{right}[v])$

 If $y < v.y_{\max}$

$v.y_{\max} \leftarrow \max\{v.y, y\}$

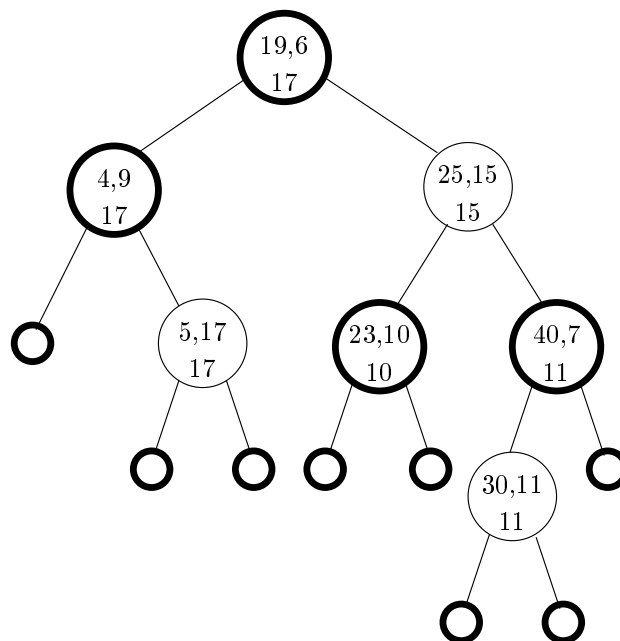
S0pdater($\text{parent}[v]$)

```

maxy( $u, v$ )
If  $u \neq \text{NIL}$ 
    if  $v \neq \text{NIL}$ 
        if  $u.y_{\max} > v.y_{\max}$ 
            return  $u.y_{\max}$ 
        return  $v.y_{\max}$ 
    return  $u.y_{\max}$ 
return 0

```

Spørgsmål c:



Det er ikke nødvendigt med omfarvninger eller rotationer, da den nye knudes forælder er sort.

Spørgsmål d: Søg så langt til venstre i træet som muligt. Dvs. brug flg. algoritme, hvor T er det udvidede rød-sortede træ.

```

MinAbove( $t$ )
 $r \leftarrow \text{root}[T]$ 
If  $r \neq \text{NIL}$  and  $r.u_{\max} \geq t$ 
     $u \leftarrow \text{MA}(r, t)$ 
    return  $(u.x, u.y)$ 
Else
    return NIL

```

```

MA( $v, t$ )
 $\ell \leftarrow \text{left}[v]$ 
if  $\ell \neq \text{NIL}$ 
    if  $\ell.y_{\max} \geq t$ 
        return MA( $\ell, t$ )
if  $v.y \geq t$ 
    return  $v$ 
return MA( $\text{right}[v], t$ )

```


Opgave 4

Spørgsmål a: $X = \langle 2, 6, 8, 10, 14, 15, 20, 21 \rangle$ og $W = 7$.

$$U(k) = \min_{1 \leq i \leq k-1} \{U(i) + (X(k) - X(i) - W)^2\}, \text{ for } k > 1.$$

k	1	2	3	4	5	6	7	8
$X[k]$	2	6	8	10	14	15	20	21
$U(k)$	0	9	1	1	2	1	3	2

Spørgsmål b: $\langle 2, 8, 14, 21 \rangle$

Spørgsmål c:

$n \leftarrow |X|$

For $k \leftarrow 2$ to n

$U(k) \leftarrow (X[k] - X[1] - W)^2$

For $i \leftarrow 2$ to $k - 1$

$\text{tmp} \leftarrow U(i) + (X(k) - X(i) - W)^2$

If $\text{tmp} < U(k)$

$U(k) \leftarrow \text{tmp}$

Return $U(n)$

Det tager tid $O(k)$ at beregne $U(k)$. Dvs. køretiden er

$$O\left(1 + \sum_{k=2}^n k\right) = O(n^2).$$

Pladsforbruget er $O(n)$, da den plads, der bruges ud over tabellen, er konstant.