Skriftlig Eksamen
Algoritmer og Datastrukturer (DM02)

Institut for Matematik og Datalogi
Syddansk Universitet, Odense

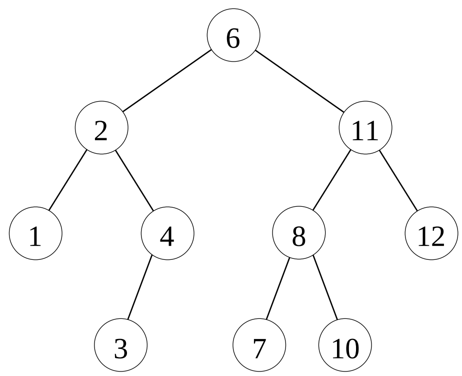
Torsdag den 11. januar 2007, kl. 9–13

Løsningsforslag

Opgave 1

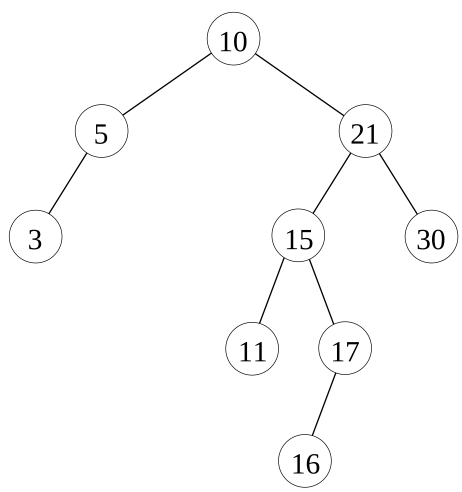
Spørgsmål a: Ja: for enhver knude x gælder, at nøglerne i x's venstre undertræ er mindre end x's nøgle, og nøglerne i højre undertræ er større. □

Spørgsmål b: Efterfølgeren til 5-knuden, dvs. 6-knuden, slettes og indsættes på 5-knudens plads:



□

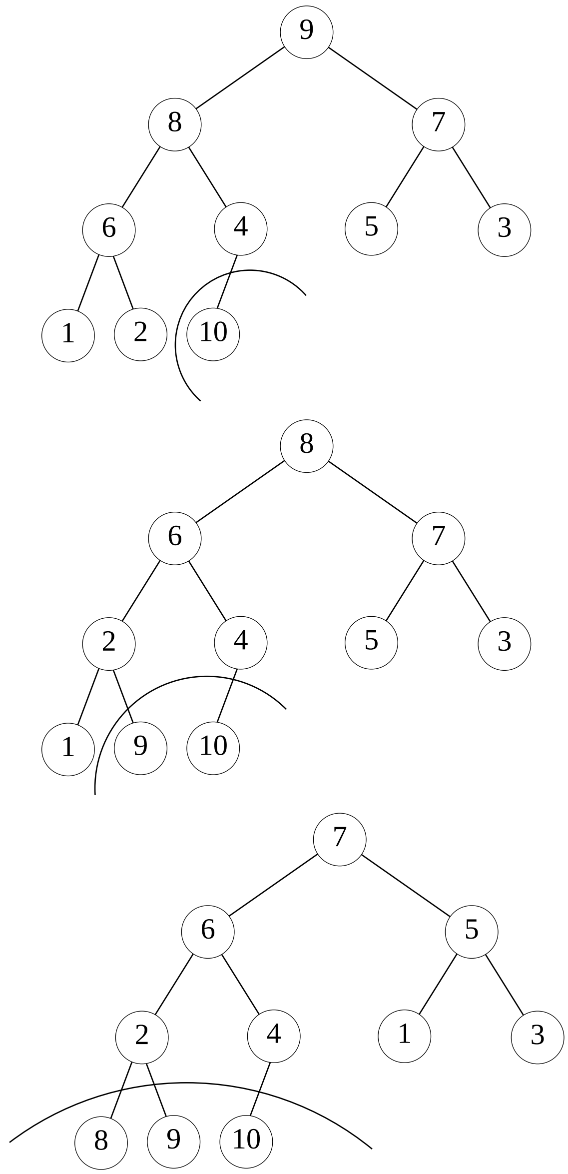
Spørgsmål c:



□

Opgave 2

Spørgsmål a:



□

Opgave 3

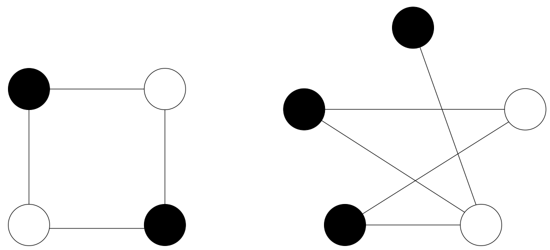
Spørgsmål a:

1. $n^2 \in \Omega(n)$ — sand
2. $n \in \Theta(n^2)$ — falsk
3. $n \log n \in o(n^2)$ — sand
4. $\log n \in O(\sqrt{n})$ — sand
5. $n! \in \omega(2^n)$ — sand

$\square$

Opgave 4

Spørgsmål a: (b) og (c) er todelte:



□

Spørgsmål b: Med udgangspunkt i pseudokoden på s. 532 i lærebogen foretages følgende ændringer:

- Mellem linie 7 og 8 tilføjes
  set$[s] \leftarrow X$
- Mellem linie 16 og 17 tilføjes
  if set$[u] = X$
    set$[v] \leftarrow Y$
  else
    set$[v] \leftarrow X$
- Mellem linie 17 og 18 tilføjes
  else if set$[v] =$ set$[u]$
    return "not bipartite"
- Efter linie 18 tilføjes
  return "bipartite"

Algoritmen er korrekt: Algoritmen skal undersøge, om knuderne kan deles i to mængder X og Y, så alle kanter går mellem X og Y. Det er lige meget, om den første knude s anbringes i X eller Y, så det er OK at vælge X. Herefter *skal* alle s's naboer være i Y. Alle deres naboer skal nødvendigvis være i X osv. Dvs. vi gør aldrig noget galt, for vi har aldrig nogen valgmulighed. Derfor er det tegn på, at grafen ikke er todelt, hvis vi på et tidspunkt opdager en kant, hvis endepunkter er i samme mængde.

Man kan også blot gøre rede for, at løsningen fra opgave 22.2-6 kan bruges. □

Spørgsmål c: "Kun hvis": I en kreds skal hver anden knude tilhøre den ene mængde, og hveranden knude den anden mængde. I en ulige kreds kan det ikke lade sig gøre.

"Hvis": Hvis algoritmen fra b) fejler, har den opdaget en ulige kreds:

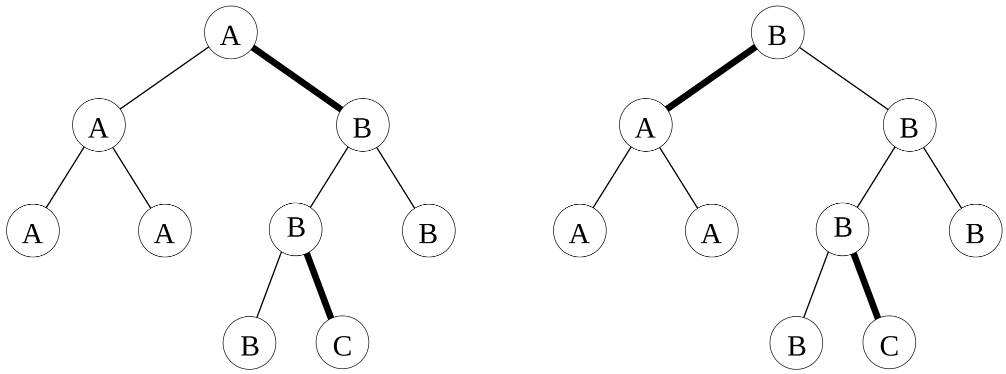
— Når den opdager en knude, som ikke er hvid, har den opdaget en kreds.

— Knuderne i denne kreds er blevet tildelt X og Y skiftevis — bortset fra de to naboknuder u og v, som den lige har opdaget; de er blevet tildelt samme mængde. Dermed har kredsen et ulige antal knuder.

Dvs. hvis grafen ikke indeholder en ulige kreds, fejler algoritmen ikke. □

Opgave 5

Spørgsmål a: Der er to optimale løsninger:



Der kan ikke være nogen bedre løsning, for antallet af overgangskanter må nødvendigvis være mindst antallet af forskellige tegn minus 1.

□

Spørgsmål b: COST kaldes 3 gange med x_1. Hvert af disse kald kalder COST med x_2 3 gange. Dvs. COST kaldes 9 gange med x_2. Hvert af disse kald kalder COST med x_4 3 gange. Altså kaldes COST ialt 27 gange med x_4. □

Spørgsmål c: For en knude x i dybde i kaldes COST $|\Sigma|^{i+1}$ gange med x som det ene argument. Et træ med n knuder har dybde mindst $\log_2 n$. □

Spørgsmål d: Opret en 2-dimensionel tabel T, hvor $T[i, a]$ angiver det mindste antal overgangskanter, man kan nøjes med i x_i's undertræ, hvis x_i indeholder tegnet a.

Der tilføjes lidt ekstra til TOTALCOST og COST:

$\underline{\text{TotalCost}}(r)$

For $i = 1$ til n
    For hvert $a \in \Sigma$
        $T[i, a] \leftarrow -1$
Returner $\min_{a \in \Sigma} \{\text{Cost}(r, a)\}$

$\underline{\text{Cost}}(k, a)$

Hvis k er et blad
    Hvis k.tegn $= a$
        returner 0
    Ellers
        returner ∞
Ellers
    Hvis $T[k.\text{id}, a] = -1$
        minLeft $\leftarrow \min_{b \in \Sigma} \{\text{Cost}(k.\text{left}, b) + \text{Overgang}(a, b)\}$
        minRight $\leftarrow \min_{b \in \Sigma} \{\text{Cost}(k.\text{right}, b) + \text{Overgang}(a, b)\}$
        $T[x.\text{id}, a] \leftarrow \text{minLeft} + \text{minRight}$
Returner $T[x.\text{id}, a]$

$\square$