Introduktion til algoritmik og datastrukturer

IT-højskolen i København

17. juni 2002

Dette eksamenssæt består af 3 opgaver med i alt 13 delopgaver. De 13 delopgaver vægtes ens i bedømmelsen. Du har i alt 4 timer til din rådighed. Husk at angive sidetal, navn og cpr.-nummer på alle sider i din besvarelse. Eksamenssættet består af 6 nummererede sider.

I opgaver, hvor der skal angives algoritmer, lægges der i bedømmelsen vægt på den beskrevne løsnings asymptotiske tidskompleksitet. I opgaver, hvor der skal angives tidskompleksitet, skal denne udtrykkes i O-notation. Der lægges vægt på, at størrelsesordener i O-notation udtrykkes med mindst mulig vækstrate.

Opgave 1

Denne opgave handler om rodfæstede, binære træer. Opgaven lægger sig op ad repræsentation og notation for binære træer, som angivet i afsnit 10.4 side 214-215 i CLRS.

Betragt følgende procedure, der som argument tager roden x for et binært træ. Vi antager, at alle knuder x har et felt $size[x]$, der indeholder et heltal.

ZERO(x)

if $x \neq$ NIL

  then $size[x] \leftarrow 0$

    ZERO($right[x]$)

    ZERO($left[x]$)

Ideen i ovennævnte procedure er, at efter udførelsen vil alle felterne $size[x]$ være 0.

a) Angiv tiden for udførelse af proceduren ZERO(x) i O-notation, hvor x er rod i et træ med n knuder.

I de følgende delopgaver betegner $T(x)$ *undertræet* med rod x i træet T (jvf. CLRS side 1087, hvor et undertræ med rod x er "subtree rooted at x") Antag, at vi ønsker en procedure INITSIZE(x), der givet roden x i et træ T med n knuder initialiserer felterne $size[y]$ for alle knuder y i T, så $size[y]$ bliver lig med antallet af knuder i $T(y)$.

b) Giv pseudokode for INITSIZE(x), så udførelsestiden er $O(n)$.

For et træ T siger vi, at en kant (u, v), hvor u er far til v, er *grøn* hvis antallet af knuder i undertræet $T(u)$ er mindst dobbelt så stort som i undertræet $T(v)$. D.v.s. efter udførelse af INITSIZE gælder $size[u] \geq 2size[v]$ for alle grønne kanter (u, v).

c) Lav en procedure, der beregner antallet af grønne kanter i hele træet T. Angiv tidskompleksiteten og argumenter for denne samt korrektheden af din algoritme.

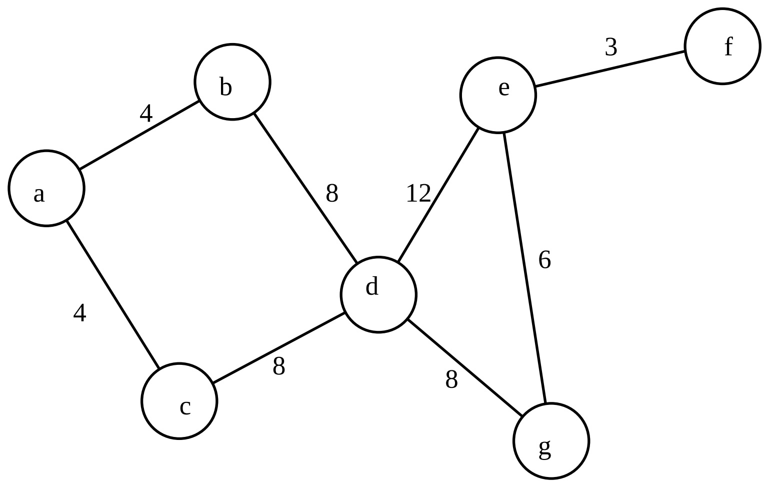
d) Angiv i O-notation en øvre grænse for antallet af grønne kanter, der er på stien fra en knude til roden af T. Argumenter for dit svar.

Betragt følgende procedure.

```
GreenPathSum(x)
if x ≠ NIL and left[x] ≠ NIL and right[x] ≠ NIL
   then for i ← 1 to size[x]
             do timecount ← timecount + 1
        if size[x] ≥ 2size[left[x]]
             then GreenPathSum(left[x])
             else GreenPathSum(right[x])
```

e) Angiv tidskompleksiteten af GreenPathSum(x). Argumenter for dit svar.

Opgave 2



Figur 1: Grafen H

a) Angiv kanterne i et mindst udspændende træ for grafen H i figur 1.

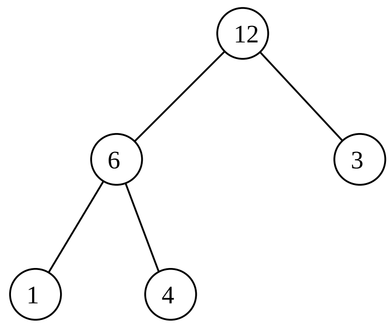
b) Angiv adjacency-list repræsentation for H i stil med figur 22.1 side 528 i CLRS.

Lad G være en sammenhængende, uorienteret graf, hvor kantvægtene $w(u, v)$ er defineret til at være produktet af graderne (*degree* som defineret på side 1081 i CLRS) for de to endeknuder u og v, d.v.s. $w(u, v) = \deg(v) \cdot \deg(u)$, hvor $\deg(u)$ og $\deg(v)$ betegner graden af henholdsvis knude u og knude v. I grafen H i figur 1 er kantvægtene netop defineret på denne vis.

c) Angiv en effektiv algoritme for at beregne mindst udspændende træ for grafer, hvor kantvægtene er som defineret for G.

d) Giv en algoritme, der givet en *orienteret* graf G med positive kantvægte og en knude v, finder den korteste simple cykel (d.v.s. *simple cycle* jvf. CLRS side 1081), hvor knuden v er med på denne cykel. Hvis der ikke findes en sådan cykel skal algoritmen returnere 0. Angiv tidskompleksiteten, og argumenter for korrektheden af din algoritme.

Opgave 3



Figur 2: Hoben B

I figur 2 er vist en hob med fem elementer med værdierne 1, 3, 4, 6 og 12.

a) Tegn hoben B i figur 2 efter udførelse af en HEAP-EXTRACT-MAX operation.

De følgende delopgaver b) - d) handler om udvidelser af repetoiret af operationer for hoben udover de i CLRS på side 129 beskrevne operationer. Disse eksisterende operationer skal hoben fortsat understøtte i de angivne tider på side 129 i CLRS. I det følgende antager vi, at elementernes værdier er positive reelle tal som vi kan foretage addition og division på i konstant tid.

Betragt nu følgende operation.

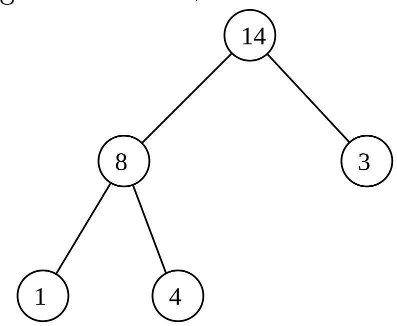
FUSION (A, x, y): Fjerner fra hoben A elementerne x og y, (d.v.s. med værdierne $A[x]$ og $A[y]$), og tilføjer et nyt element, hvis værdi er $A[x] + A[y]$.

b) Angiv en algoritme for operationen FUSION. Argumenter for køretiden.

Betragt følgende nye hob-operation.

ADDUPTO (A, q, v): Lægger den positive værdi v til alle elementer y i hoben A, hvor $A[y] \geq q$.

F.eks. vil hoben B i figur 2 efter udførelse af ADDUPTO$(B, 6, 2)$ ændre sig til:

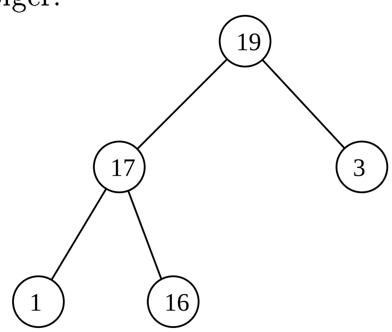


c) Beskriv en algoritme for operationen ADDUPTO så tiden bliver $O(k)$, hvor k er antallet af elementer, der ændrer værdi ved udførelse af operationen.

Betragt følgende operation:

ADDFUNNY(A, k): Lægger værdien $\frac{60}{A[x]}$ til værdien for element x, hvis x er blandt de k største elementer i hoben A.

F.eks. vil hoben B i figur 2 efter udførelse af ADDFUNNY(3) indeholde fem elementer med værdierne 1, 3, 16, 17 og 19, da de tre hidtil største værdier 12, 6 og 4 bliver til henholdsvis 17, 16 og 19. Hoben kunne efter udførelse af denne operation så se ud som følger:



d) Beskriv en algoritme for operationen ADDFUNNY(k), så tiden bliver $O(k \lg k)$ for denne, mens de øvrige tider bibeholdes.