DATALOGISK INSTITUT, AARHUS UNIVERSITET

Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 6 (seks)
Eksamensdag: Fredag den 20. august 2010, kl. 9.00-13.00
Eksamenslokale: Åbogade 34, Benjaminbygningen, indgang B, 8200 Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger, notater, lommeregner). Computer må ikke medbringes.
Materiale der udleveres til eksaminanden:

Oplysninger til eksamensadministrationen
vedrørende opgavepakke fra **Datalogisk Institut**

dato: kl.

Antal indlagte opgaver **18**

Evt. faglærer der vil være til stede ved eksamens begyndelse _____

Navn og tlf. på "vagthavende faglærer" under eksamen **Gerth Brodal 5059 5432 / Mark Greve 6075 7568**

Opgavebesvarelserne:
- [x] 1) afhentes i eksamenslokalet ved eksamens afslutning
- [ ] 2) afhentes i administrationen den følgende morgen
- [ ] 3) ønskes sendt til _____

Navn på den/de person(er), der er bemyndiget til at afhente/modtage opgavebesvarelserne _____ Mark Greve

Forbeholdt eksamensadministrationen:

Opgave modtaget:

DATALOGISK INSTITUT, AARHUS UNIVERSITET

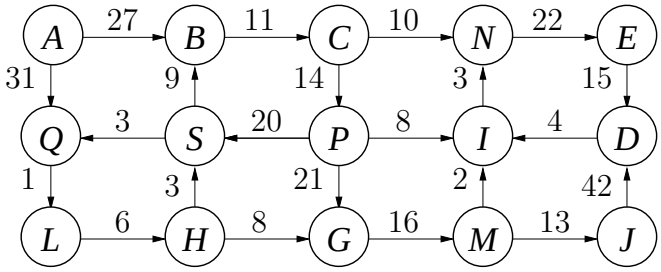
Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 6 (seks)
Eksamensdag: Fredag den 20. august 2010, kl. 9.00-13.00
Eksamenslokale: Åbogade 34, Benjaminbygningen, indgang B, 8200 Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger, notater, lommeregner). Computer må ikke medbringes.
Materiale der udleveres til eksaminanden:

OPGAVETEKSTEN
BEGYNDER
PÅ NÆSTE SIDE

—oOo—

Opgave 1 (25%)

I spørgsmål a-c betragter vi nedenstående orienterede graf. Det antages, at grafen er givet ved incidenslister, hvor incidenslisterne er sorteret alfabetisk.

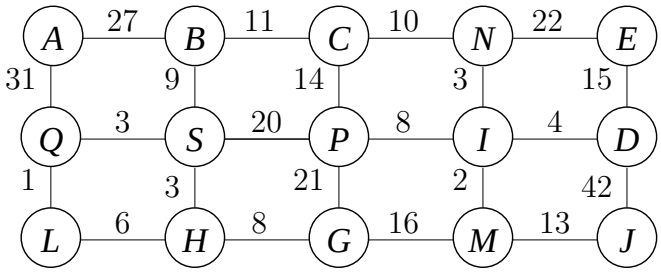


Spørgsmål a: Angiv et BFS-træ for ovenstående graf, når BFS-gennemløbet starter i knuden A. Angiv kanterne i BFS-træet og BFS-numrene for knuderne. □

Spørgsmål b: Angiv et DFS-træ for ovenstående graf, når DFS-gennemløbet starter i knuden A, og en DFS-nummerering af knuderne. Angiv for hver knude "discovery time" og "finishing time". □

Spørgsmål c: Angiv et korteste veje træ for ovenstående graf, når kortest veje beregningen sker med hensyn til startknuden A. For hver knude v angiv også afstanden fra startknuden A til v. □

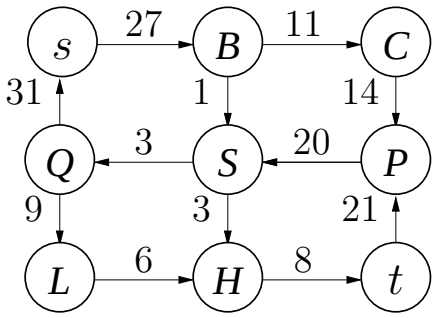
Spørgsmål d: Angiv de stærke sammenhængskomponenter i ovenstående graf. □



Spørgsmål e: Angiv kanterne i et minimum udspændende træ for ovenstående uorienterede graf med vægte på kanterne. □

Opgave 2 (15%)

Betragt nedenstående netværk med de angivne kapaciteter på kanterne.



Spørgsmål a: Angiv en maksimal strømning fra s til t i netværket (angiv for hver kant strømningen langs kanten), angiv værdien af en maksimal strømning, og angiv et snit (dvs. opdeling af knuderne i to disjunkte mængder) hvor kapaciteten af snittet er lig værdien af en maksimal strømning. □

Spørgsmål b: Betragt Edmonds-Karp algoritmen anvendt på ovenstående graf til beregning af en maksimal strømning. Angiv de forbedrende stier der anvendes under udførslen af Edmonds-Karp algoritmen. For hver forbedrende sti angiv knuderne på stien og strømningen, man forbedrer med, langs stien. □

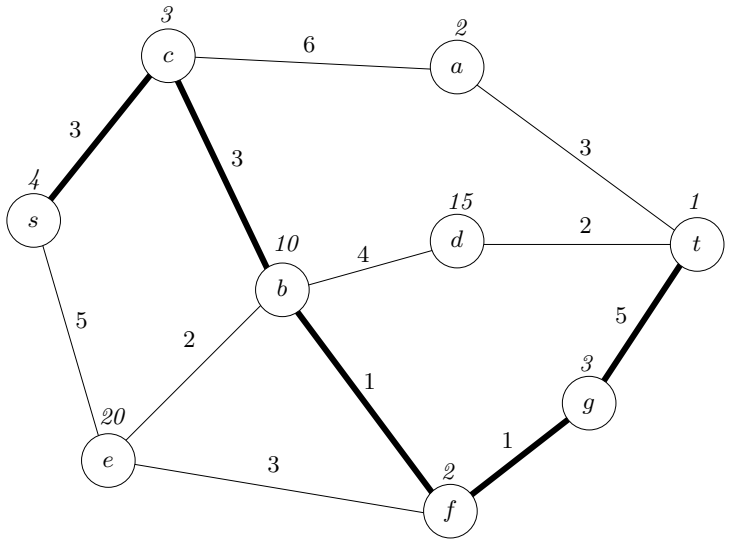
(Opgavesættet fortsætter)

Opgave 3 (20%)

I denne opgave betragter vi en uorienteret graf $G = (V, E)$, repræsenterende et vejnet, hvor $n = |V|$ og $m = |E|$. Hver kant (u, v) har en heltallig længde $w(u, v)$, der angiver hvor mange liter benzin man skal bruge for at køre mellem u og v. I hver knude u er der en tankstation, hvor man kan købe benzin til literprisen $p(u)$ (angivet skraveret over knuderne i nedenstående eksempel), hvor $p(u)$ er et positivt heltal. Det antages, at bilens tank højst kan indeholde C liter benzin, og at man kun kan tanke et heltalligt antal liter.

I denne opgave ønsker vi at finde en vej fra en knude s til en knude t i grafen, hvor bilen ikke løber tør for benzin, mens man kører langs en kant. Tanken starter med at være tom i s.

Eksempel: I nedenstående eksempel antager vi, at tanken har størrelse $C = 5$. Den fede sti angiver en sti fra s til t hvor der skal bruges $3+3+1+1+5 = 13$ liter benzin. Ved at tanke henholdsvis 3, 4, 0, 5, og 1 liter ved tankene s, c, b, f, og g, bliver prisen for at komme fra s til t totalt $3 \cdot \mathit{4} + 4 \cdot \mathit{3} + 0 \cdot \mathit{10} + 5 \cdot \mathit{2} + 1 \cdot \mathit{3} = 37$ (og på intet tidspunkt er der mere end 5 liter benzin i tanken efter der er tanket).



Spørgsmål a: Beskriv en algoritme, der afgør om man kan komme fra s til t med en bil med tankstørrelse C. Angiv algoritmens udførselstid. □

Spørgsmål b: Under antagelse at man kan komme fra s til t uden at løbe tør for benzin, beskriv en algoritme, der finder den billigste pris man kan komme fra s til t. Angiv algoritmens udførselstid, som funktion af n, m og C. Hint: Lav en vægtet orienteret graf med knuderne $\langle u, t \rangle$, hvor u er en knude i den oprindelige graf og t er antal liter benzin i tanken. □

(Opgavesættet fortsætter)

Opgave 4 (20%)

I denne opgave antager vi, at vi er givet k identiske glaskugler, og ønsker at bestemme den laveste højde højde glaskuglerne går i stykker fra, hvis vi lader dem falde. Der findes en heltallig højde (etage) H, hvor alle glaskuglerne går i stykker, hvis vi lader dem falde fra en højde $\geq H$ men ikke for højde $< H$. Vi antager, at vi kan lade kuglerne falde fra n forskellige etager i et hus med etagerne $1, 2, \ldots, n$, og at $1 \leq H \leq n+1$. I denne opgave ønsker vi at finde en algoritme til at bestemme det mindste antal gange, vi skal lade en glaskugle falde for at bestemme H.

Eksempel: Hvis $k = 1$, så er den eneste mulighed for at bestemme H ved at kaste glaskuglen ud fra hver af etagerne $1, 2, 3, \ldots$ indtil glaskuglen går i stykker, hvilket sker ved etage H. Dette kræver H kast, hvilket i værste tilfælde er n. Hvis $k = 2$, kan vi f.eks. bestemme H ved at lade den første kugle falde fra højderne $2, 4, 6, 8, \ldots$ indtil vi når en højde $2i$ hvor kuglen går i stykker eller $2i > n$. Vi ved nu at $2(i-1) < H \leq 2i$. Ved at kaste den anden kugle kugle fra højde $2i - 1$ og ser om den går i stykker kender vi nu H. Dette kræver totalt højst $n/2 + 1$ kast (for de fleste n kan man dog gøre det væsentligt bedre med to glaskugler).

For $1 \leq i \leq k$ og $0 \leq h \leq n$ lader vi $B(i, h)$ angive, hvor mange kast der skal til for at bestemme H blandt $h + 1$ forskellige højder med i glaskugler. $B(i, h)$ kan beskrives ved følgende rekursionsformel:

$$B(i,h) = \begin{cases} 0 & \text{hvis } h = 0 \\ h & \text{hvis } h > 0 \ \wedge \ i = 1 \\ 1 + \min_{j \in \{1,..,h\}} \max\{B(i-1, j-1), B(i, h-j)\} & \text{hvis } h > 0 \ \wedge \ i \geq 2 \end{cases}$$

Spørgsmål a: Udfyld nedenstående tabel for $B(i, h)$ for $k = 2$ glaskugler og $n = 6$ etager.

$i \setminus h$	0	1	2	3	4	5	6
1							
2							

□

Spørgsmål b: Angiv en algoritme baseret på dynamisk programmering der, givet k og n, bestemmer det mindste antal kast for at bestemme H. Angiv algoritmens udførselstid. □

Spørgsmål c: Angiv en algoritme der, givet k og n, bestemmer en etage for det første kast med den første glaskugle, hvis man skal opnå det mindste antal kast for at bestemme H. □

(Opgavesættet fortsætter)

Opgave 5 (20%)

I det følgende angiver $S = (x_1, x_2, \ldots, x_n)$ en sekvens af n heltal. Vi ønsker at finde en længste sammenhængende delsekvens $S[i : j] = (x_i, x_{i+1}, \ldots, x_j)$, hvor alle tal er forskellige.

Eksempel: Sekvensen $S = (4, \underline{3, 7, 5, 2, 4, 6}, 5, 3)$ har delsekvensen $S[2 : 7] = (3, 7, 5, 2, 4, 6)$ som en længste sammenhængende delsekvens, hvor alle tal er forskellige.

Spørgsmål a: Angiv for følgende sekvens en længste sammenhængende delsekvens, hvor alle tal i delsekvensen er forskellige.

$$S = (4, 3, 8, 1, 7, 5, 4, 2, 8, 9, 4, 7, 6, 2, 7, 6, 3, 4, 2, 1)$$

□

Spørgsmål b: Beskriv en algoritme, der givet en sekvens af n tal, finder den længste sammenhængende delsekvens, hvor alle tal i delsekvensen er forskellige. Angiv algoritmens udførselstid. □

Spørgsmål c: Beskriv en algoritme med udførselstid $O(n \log n)$, der finder den længste sammenhængende delsekvens af en sekvens af n tal, hvor hvert tal højst indgår **to gange** i sekvensen. □