DATALOGISK INSTITUT, AARHUS UNIVERSITET

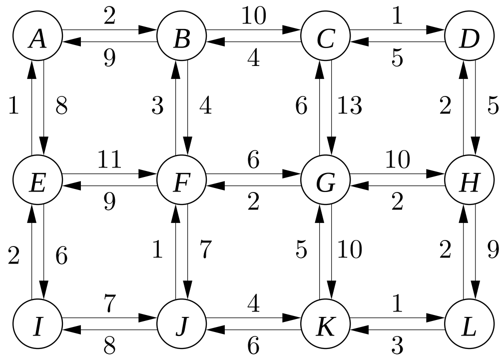
Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 6 (seks)
Eksamensdag: Fredag den 24. juni 2011, kl. 9.00-13.00
Eksamenslokale: Trøjborg, Willemoesgade 15, Bygn 2113, Lok 139
Tilladte medbragte hjælpemidler: 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 denne opgave betragter vi først orienterede *gitter-grafer* med ikke-negative vægte på kanterne. En gitter-graf består af M rækker hver med N knuder. Det antages, at grafen er givet ved *incidenslister*, hvor incidenslisterne er sorteret alfabetisk.

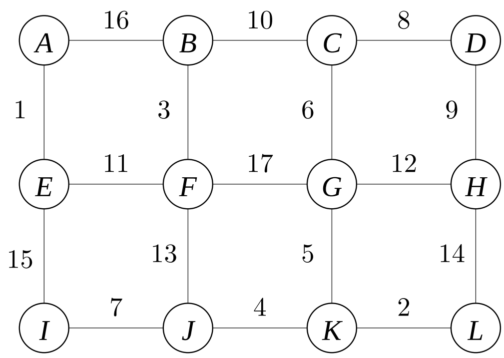


Spørgsmål a: Angiv et BFS-træ for ovenstående graf, hvor BFS-gennemløbet starter i knuden A. Angiv kanterne i BFS-træet og BFS-numrene for knuderne. □

Spørgsmål b: Angiv et DFS-træ for ovenstående graf, hvor 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 (SSSP) 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. □

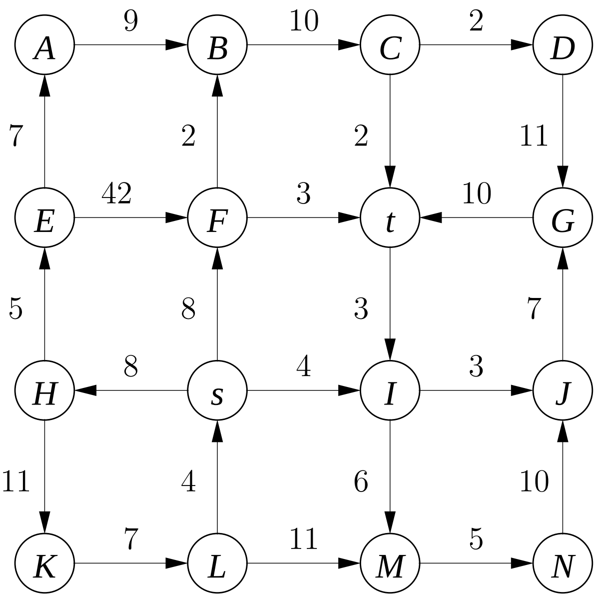
Spørgsmål d: For en gitter-graf med $M \cdot N$ knuder, angiv en algoritme til at beregne *diameteren* af grafen, dvs. den maksimale afstand mellem to knuder i grafen, dvs. afstanden $\max\{d(u,v) \mid u,v \in V\}$, hvor $d(u,v)$ angiver længden af den korteste vej mellem knuderne u og v. Angiv algoritmens udførselstid. □



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

Opgave 2 (15%)

Betragt nedenstående netværk med de angivne kapaciteter på kanterne.



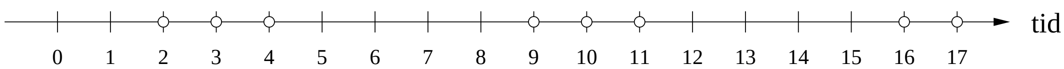
Spørgsmål a: Angiv en maksimal strømning (maximum flow) fra s til t i netværket (angiv for hver kant strømningen langs kanten), angiv værdien af en maksimal strømning, og angiv et snit (cut) (dvs. opdeling af knuderne i to disjunkte mængder) hvor kapaciteten af snittet er lig værdien af en maksimal strømning.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

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

Opgave 3 (20%)

I denne opgave betragter vi et vejnet modeleret ved en orienteret graf, en *vejgraf*, og ønsker at finde den hurtigste vej fra en knude S til en knude T. For hver kant (u, v) har vi en positiv vægt $w(u, v)$, som angiver antal sekunder det tager at køre langs kanten (u, v). Hver knude v angiver et lyssignal, som skiftevis er grøn og rød. Første gang lyssignalet bliver grønt er til tiden $\Delta_0(v)$, og derefter skiftes det til at være grønt og rødt i henholdsvis $\Delta_{\text{grøn}}(v)$ og $\Delta_{\text{rød}}(v)$ sekunder. Det antages at $0 \leq \Delta_0(v) \leq \Delta_{\text{rød}}(v)$. Det antages at alle tidspunkter er heltallige.

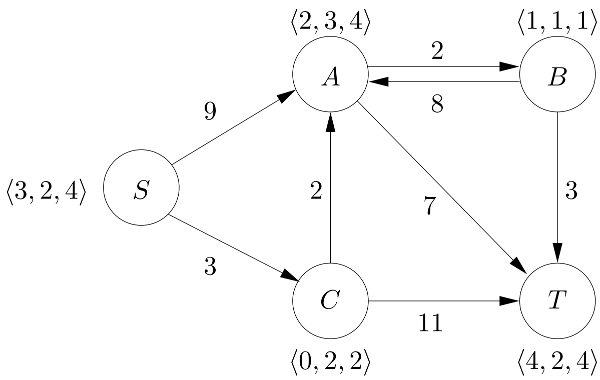
Nedenstående illustrerer hvornår der er grønt når $\Delta_0(v) = 2$, $\Delta_{\text{grøn}}(v) = 3$, og $\Delta_{\text{rød}}(v) = 4$. Tidspunkter med grønt er angivet med cirkler.



Med **release**(v, t) angiver vi det første tidspunkt $\geq t$, hvor det bliver grønt i knuden v. For ovenstående eksempel er **release**$(v, 10) = 10$ og **release**$(v, 12) = 16$.

Spørgsmål a: Angiv en algoritme for hvordan **release**(v, t) kan beregnes. □

Betragt nedenstående vejgraf. For hver kant (u, v) er angivet $w(u, v)$ og for hver knude er angivet $\langle \Delta_0(v), \Delta_{\text{grøn}}(v), \Delta_{\text{rød}}(v) \rangle$. Betragt stien $S \to A \to T$. Når vi starter i S til tiden 0, kan vi forlade S til tiden 3, da dette er første gang der er grønt ved S. Vi ankommer så til A til tiden 3+9=12. Da det bliver grønt ved A til tiden 16, som angivet i ovenstående eksempel, kan vi forlade A og ankomme til T til tiden 16+7=23.



Spørgsmål b: Angiv for hver knude i ovenstående vejgraf det tidligste tidspunkt man kan ankomme til en knude og forlade en knude, når man starter ved S til tiden 0.

Knude	:	S	A	B	C	T
Tidligste ankomst	:	0				
Tidligste afgang	:	3				–

□

Spørgsmål c: Beskriv en algoritme, der givet en vejgraf med n knuder og m kanter, finder det tidligste tidspunkt man kan komme til en knude T, når man starter i knuden S til tiden 0. Angiv algoritmens udførselstid. □

(Opgavesættet fortsætter)

Opgave 4 (20%)

I denne opgave ønsker vi at finde det mindste antal mønter der skal til for at opnå et givent beløb n, når mønterne antager værdier i $X = \{x_1, x_2, ..., x_k\}$, dvs. vi har k forskellige slags mønter. Vi lader C_i angive det færreste antal mønter der skal bruges for at opnå værdi i. F.eks. hvis mønterne antager værdierne $X = \{1, 5, 8\}$ kan vi opnå $i = 26$ med fire mønter, da $26 = 5 + 5 + 8 + 8$, dvs. $C_{26} = 4$. Vi antager i det følgende at $1 \in X$ således at alle beløb er garanteret at kunne angives ved hjælp af mønterne.

Antallet C_i kan bestemmes ved følgende rekursionsformel:

$$C_i = \begin{cases} 0 & \text{hvis } i = 0 \\ 1 + \min\{C_{i-x_j} \mid 1 \le j \le k \wedge x_j \le i\} & \text{ellers} \end{cases}$$

Spørgsmål a: Udfyld nedenstående tabel for $X = \{1, 3, 7\}$ og $0 \le i \le 25$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C_i :																										

□

Spørgsmål b: Angiv en algoritme baseret på dynamisk programmering, der givet en mængde X og et tal n beregner C_n. Angiv algoritmens udførselstid. □

Spørgsmål c: Udvid algoritmen fra spørgsmål b) til også at rapportere hvilke mønter, der skal anvendes for at opnå beløbet n. Angiv algoritmens udførselstid. □

(Opgavesættet fortsætter)

Opgave 5 (20%)

Antag at vi er givet k strenge $I_1, \ldots, I_k$, som vides at være *virus*-befængte, og en streng T som er virus-fri. Den samlede længde af strengene er $n = |T| + \sum_{i=1}^{k} |I_i|$. Vi ønsker at finde en kortest mulig *indikatorstreng* V, dvs. en streng der er en delstreng af alle $I_1, \ldots, I_k$ men hvor V ikke forekommer i T, eller at afgøre at der ikke findes nogen indikatorstreng.

F.eks. er V = ABCA en indikatorstreng for nedenstående mængde af strenge - den er dog ikke kortest mulig.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$I_1 =$	A	B	C	A	B	C	C	A																
$I_2 =$	C	C	A	B	C	A	A	B	A															
$I_3 =$	B	A	B	C	A	A	B	C	C	A														
$I_4 =$	C	C	A	B	C	A	C	A	B	B														
$I_5 =$	A	B	C	A	A	B	C	C	A															
$T =$	A	B	A	A	A	C	C	B	A	B	A	C	A	B	C	C	C	B	C	A	A	B	B	A

Spørgsmål a: Angiv en kortest mulig indikatorstreng V for ovenstående strenge, og angiv positionen af dens første forekomst i $I_1, \ldots, I_5$. □

I det følgende kan det antages at et suffikstræ for en streng af længde n over et alfabet med $O(1)$ tegn kan konstrueres i $O(n)$ tid.

Spørgsmål b: Beskriv en algoritme, der givet strengene $I_1, \ldots, I_k$ og T med samlet længde n, finder en kortest mulig indikatorstreng V eller rapporterer at der ingen indikatorstreng findes. Angiv algoritmens udførselstid. □