DATALOGISK INSTITUT, AARHUS UNIVERSITET

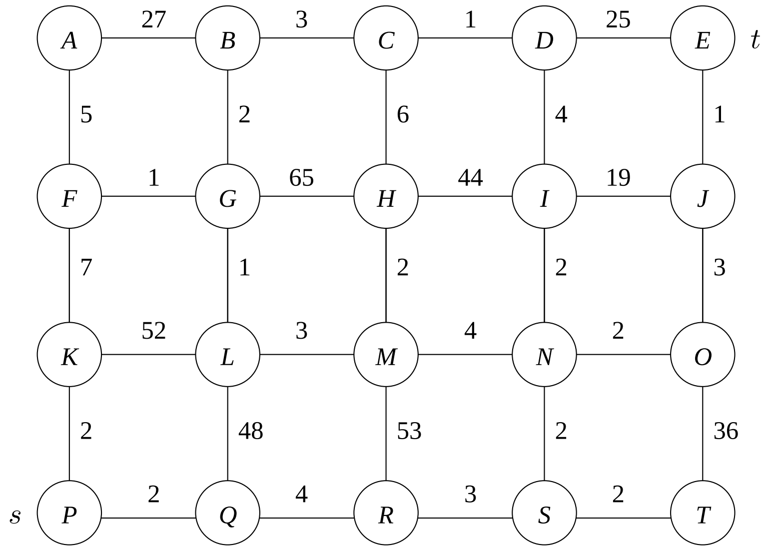
Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 7 (syv)
Eksamensdag: Torsdag den 14. juni 2007, kl. 9.00-13.00
Eksamenslokale: Skøjtehallen, Gøteborg Allé 9, Århus N
Tilladte medbragte hjælpemidler: 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 denne opgave betragtes *net-grafer*. En net-graf er en uorienteret vægtet graf hvor knuderne er arrangeret i et net af k rækker og ℓ søjler, således at en knude er forbundet med knuderne til venstre og højre for knuden (hvis disse findes) og knuderne over og under knuden (hvis disse findes). Vægtene på kanterne antages at være positive heltal. Nedenstående viser en net-graf for $k = 4$ og $\ell = 5$.



Den nederste venstre knude i en net-graf betegnes s og den øverste højre knude betegnes t.

Spørgsmål a: Angiv antal knuder n og kanter m i en net-graf som funktion af antal rækker k og søjler ℓ. Angiv som funktion af k og ℓ udførselstiden for Dijkstra's algoritme for at finde en korteste vej fra s til t i en net-graf. □

Spørgsmål b: Angiv en kortest vej fra s til t i ovenstående eksempel (hvor s og t betegner knuderne P og E). Angiv knuderne langs stien og længden af stien. □

I resten af denne opgave vil vi kun betragte stier fra s til t hvor man *ikke drejer i to knuder efter hinanden*, d.v.s. imellem to knuder hvor man drejer skal der mindst være en knude hvor man fortsætter lige ud. I ovenstående eksempel er stien

$$P - \hat{Q} - \hat{L} - M - N - \hat{O} - J - E$$

ikke en lovlig sti (en ˆ angiver at der drejes i knuden), da man drejer i både Q og L. Derimod er stien

$$P - Q - \hat{R} - M - H - \hat{C} - B - \hat{A} - F - \hat{K} - L - M - N - \hat{O} - J - E$$

en lovlig sti da der mellem alle knuderne hvor der drejes er mindst en knude hvor der køres lige ud (bemærk at knuden M besøges to gange).

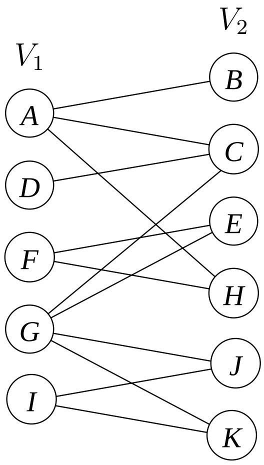
Spørgsmål c: Angiv en kortest vej fra s til t i ovenstående eksempel hvor man ikke drejer i to knuder efter hinanden. Angiv knuderne langs stien og længden af stien. □

Spørgsmål d: Beskriv en algoritme der finder den kortest vej fra s til t i en net-graf hvor man ikke drejer i to knuder umiddelbart lige efter hinanden. Angiv algoritmens udførselstid som funktion af antal rækker k og søjler ℓ. (Lav f.eks. en orienteret graf hvor hver knude i net-grafen bliver repræsenteret af flere knuder og således at en kortest sti i net-grafen hvor man ikke drejer i to knuder efter hinanden svarer til en korteste sti i den nye orienterede graf.) □

(Opgavesættet fortsætter)

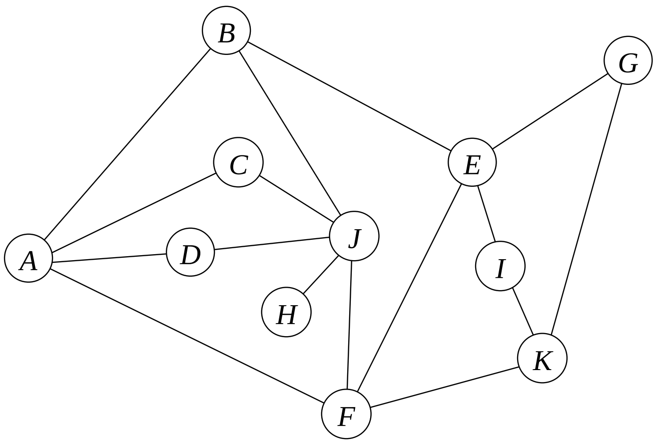
Opgave 2 (15%)

En uorienteret graf siges at være *to-delt* hvis grafens knuder V kan deles op i to mængder V_1 og V_2, således at alle kanter i G går mellem en knude i V_1 og en knude i V_2. D.v.s. der findens ingen kanter der forbinder to knuder V_1 eller to knuder i V_2. Et eksempel på en to-delt graf er følgende graf:



Bemærk at en to-delt graf kan aldrig indholde en cykel med et ulige antal kanter.

Spørgsmål a: Vis at nedenstående graf er to-delt ved at angive en opdeling af knuderne i V_1 og V_2 så alle kanter går imellem en knude i V_1 og en knude i V_2.

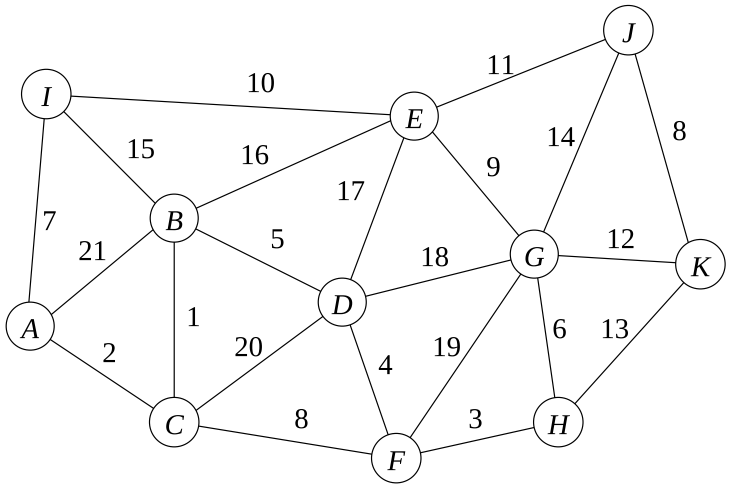


□

Spørgsmål b: Beskriv en algoritme der afgør om en sammenhængende graf med n knuder og m kanter er to-delt. Angiv algoritmens udførselstid. □

Opgave 3 (15%)

I denne opgave betragter vi minimum udspændende træer for vægtede uorienterede grafer. Det antages at alle vægte er forskellige.



Spørgsmål a: Angiv kanterne i et minimum udspændende træ for ovenstående graf og vægten af træet. □

I det følgende ønsker vi at vedligeholde et minimum udspændende træ under fjernelse af kanter i en eksisterende graf. Man kan i det følgende bruge nedenstående sætning om minimum udspændende træer uden bevis:

> (Snitsætningen) *Lad G være en vægtet uorienteret graf hvor alle kanter har forskellige vægte. Lad V_1 og V_2 være være en opdeling af knuderne i G. Et minimum udspændende træ for G indeholder den letteste kant mellem V_1 og V_2.*

Spørgsmål b: Lad G være en vægtet uorienteret graf og T et minimum udspændende træ for G. Antag vi vil *slette* en kant (u, v) fra grafen. Lad G' være den nye graf. Beskriv en algoritme til at finde et minimum udspændende træ T' for G', når det minimum udspændende træ T er kendt for G. Algoritmens udførselstid skal være $O(m)$ hvor m er antal kanter i grafen G. (Brug snitsætningen hvis (u, v) er en kant i T.) □

Opgave 4 (25%)

I denne opgave ønsker vi, givet to sekvenser $S = x_1, \ldots, x_m$ og $T = y_1, \ldots, y_n$ af længde m og n, at finde en ny sekvens U som indeholder S og T som delsekvenser. En sådan sekvens U betegnes en *super-sekvens* for sekvenserne S og T.

Hvis $S =$ A B A B A A og $T =$ B B A B B, så er nedenstående sekvenser eksempler på super-sekvenser for S og T (understregning og overstregningen viser forekomsten af S og T):

$$U_1 = \underline{\mathrm{A}}\ \underline{\mathrm{B}}\ \underline{\mathrm{A}}\ \underline{\mathrm{B}}\ \underline{\mathrm{A}}\ \underline{\mathrm{A}}\ \overline{\mathrm{B}}\ \overline{\mathrm{B}}\ \overline{\mathrm{A}}\ \overline{\mathrm{B}}\ \overline{\mathrm{B}}$$
$$U_2 = \underline{\mathrm{A}}\ \underline{\overline{\mathrm{B}}}\ \overline{\mathrm{B}}\ \underline{\overline{\mathrm{A}}}\ \overline{\mathrm{B}}\ \underline{\overline{\mathrm{B}}}\ \underline{\mathrm{A}}\ \underline{\mathrm{A}}$$
$$U_3 = \underline{\mathrm{A}}\ \underline{\overline{\mathrm{B}}}\ \underline{\mathrm{A}}\ \underline{\overline{\mathrm{B}}}\ \underline{\overline{\mathrm{A}}}\ \overline{\mathrm{B}}\ \underline{\mathrm{A}}\ \overline{\mathrm{B}}$$

Sekvenserne U_2 og U_3 er eksempler på *korteste* super-sekvenser for S og T, da alle super-sekvenser for eksemplet vil have længde 8 eller mere.

Vi lader $L(i,j)$ betegne *længden af den korteste super-sekvens* for $x_1, \ldots, x_i$ og $y_1, \ldots, y_j$. Bemærk at $L(m,n)$ er længden af den korteste super-sekvens for S og T.

$L(i,j)$ kan beskrives ved følgende rekursionsformel:

$$L(i,j) = \begin{cases} j & \text{hvis} \quad i = 0 \\ i & \text{hvis} \quad j = 0 \\ 1 + \min\{L(i-1,j), L(i,j-1)\} & \text{hvis} \quad i > 0 \land j > 0 \land x_i \neq y_j \\ 1 + L(i-1,j-1) & \text{hvis} \quad i > 0 \land j > 0 \land x_i = y_j \end{cases}$$

Spørgsmål a: Udfyld nedstående tabel for $L(i,j)$ når $S =$ A B A B og $T =$ B B A.

$L(i,j)$	0	1	2	3
0				
1				
2				
3				
4				

□

Spørgsmål b: Angiv en algoritme baseret på dynamisk programmering der finder længden af en kortest super-sekvens for to sekvenser S og T af længde m og n. Angiv algoritmens udførselstid. □

Spørgsmål c: Udvid algoritmen fra spørgsmål b) til at rapportere en kortest super-sekvens for to sekvenser S og T af længde m og n. Angiv algoritmens udførselstid. □

Opgave 5 (20%)

Spørgsmål a: Angiv suffix-arrayet for strengen: B A N A N A S . □

Spørgsmål b: Angiv suffix-træet for strengen: B A N A N A S . □

I det følgende ønsker vi givet en streng S og et heltal k at finde en delstreng i S af længde k som forekommer flest gange – forekomsterne må godt overlappe. For $k = 4$ er den mest hyppige forekomst af en streng af længde k i nedenstående eksempel delstrengen A B C A, som forekommer 6 gange.

$$S = \mathrm{A\,B\,C\,A\,B\,C\,A\,C\,B\,A\,B\,C\,A\,B\,C\,A\,A\,B\,C\,A\,C\,B\,A\,B\,C\,B\,A\,B\,C\,A\,A\,C}$$

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

Spørgsmål c: Beskriv en algoritme der givet en streng S og et heltal k finder en delstreng i S af længde k som forekommer flest gange. Det antages at S har længde n og er fra et alfabet med $O(1)$ tegn. Angiv algoritmens udførselstid. □