DATALOGISK INSTITUT, AARHUS UNIVERSITET

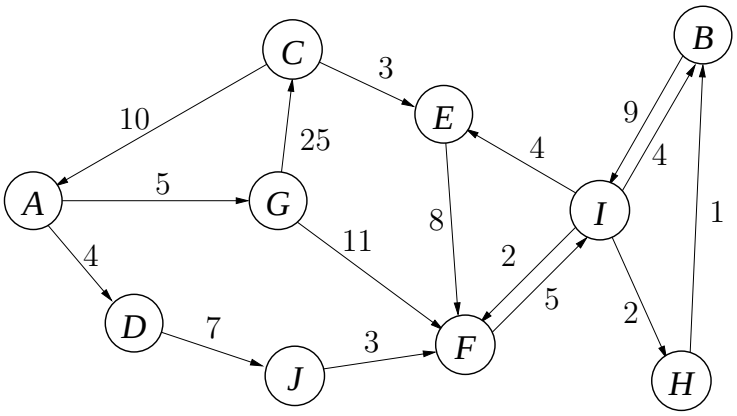
| |
|---|
| Det Naturvidenskabelige Fakultet |
| EKSAMEN |
| Grundkurser i Datalogi |
| **Algoritmer og Datastrukturer 2 (2003-ordning)** |
| Antal sider i opgavesættet (incl. forsiden): 6 (seks) |
| Eksamensdag: Mandag den 23. juni 2008, kl. 9.00-13.00 |
| Eksamenslokale: Aulaen, bygning 1412, Ndr. Ringgade, 8000 Århus C |
| Tilladte medbragte hjælpemidler:<br>Alle sædvanlige hjælpemidler (lærebøger, notater, lommeregner).<br>Computer må ikke medbringes. |
| Materiale der udleveres til eksaminanden: |

OPGAVETEKSTEN
BEGYNDER
PÅ NÆSTE SIDE

—oOo—

**Opgave 1** (25%)

Betragt nedenstående orienterede graf med ikke-negative vægte på kanterne.



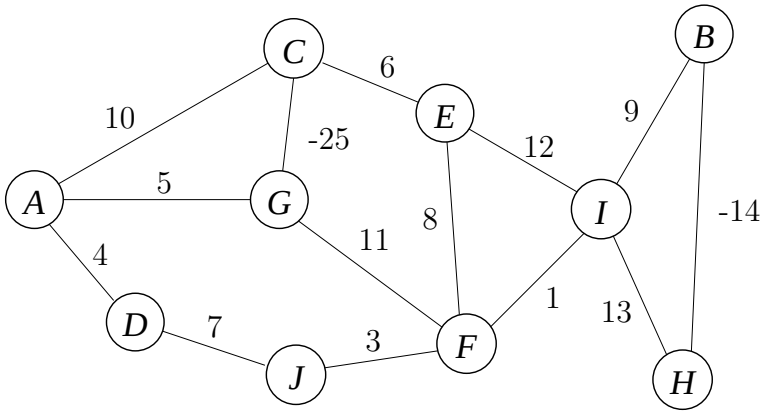

**Spørgsmål a:** Angiv et BFS træ for ovenstående graf, når BFS gennemløbet starter i knuden $A$. □

**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". Det antages at incidenslisterne er sorteret leksikografisk. □

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

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

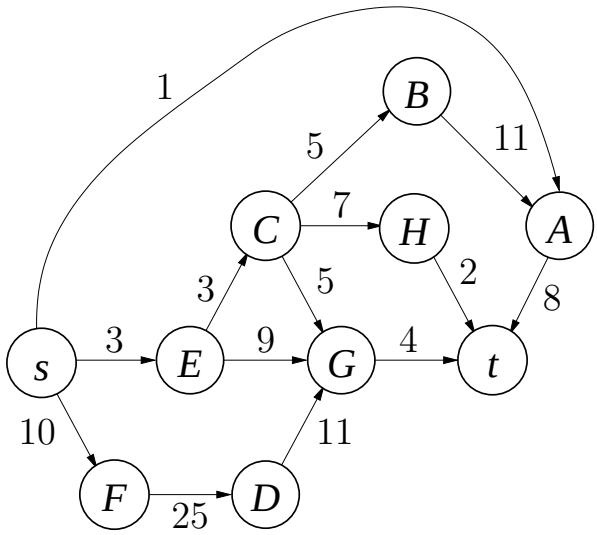
Betragt nu følgende uorienterede graf med vægte på kanterne.




**Spørgsmål e:** Angiv et minimum udspændende træ for grafen. □

**Opgave 2** (15%)

Betragt nedenstående netværk med de angive kapaciteter på kanterne.




**Spørgsmål a:** Angiv en maximal strømning fra $s$ til $t$ i netværket og et snit hvor kapaciteten af snittet er lig med den maximal strømning. □

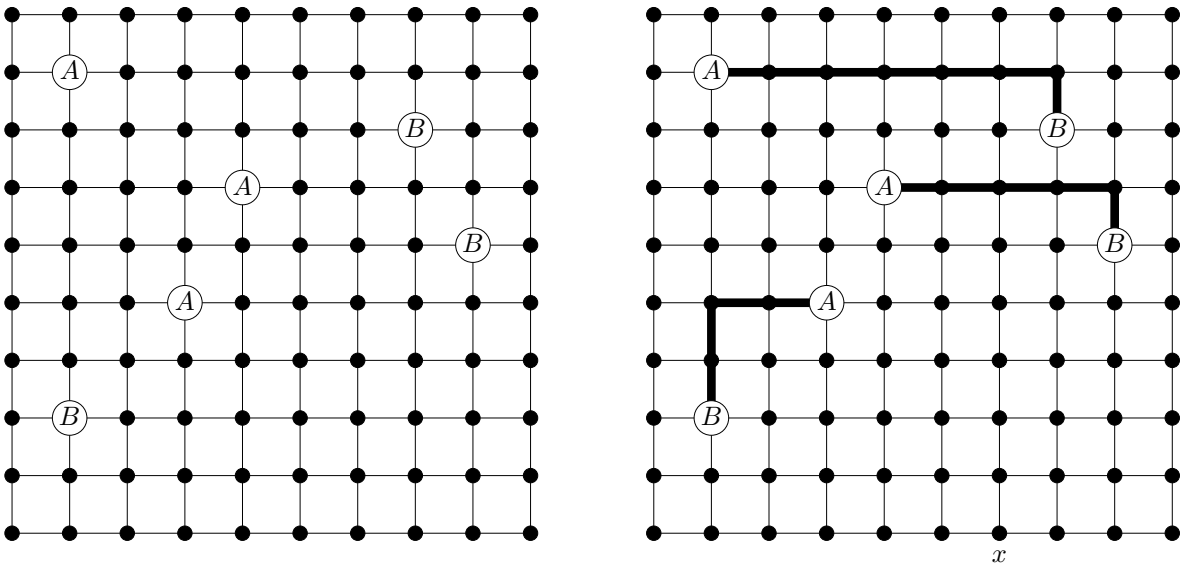
**Spørgsmål b:** Betragt Edmonds-Karp algoritmen anvendt på ovenstående graf til beregning af en maximal strømning. Angiv de forbedrende stier der anvendes under udførslen af Edmonds-Karp algoritmen. For hver forbedrende sti angiv også strømningen man forbedrer med langs stien. □

(Opgavesættet fortsætter)

**Opgave 3** (20%)

I denne opgave betragter vi kvadratiske gittergrafer, som består af et gitter med $g \times g$ knuder. Enhver knude er forbundet med sin venstre og højre nabo samt med dem umiddelbart over og under knuden. Desuden er $k$ af knuderne markeret med $A$ og yderligere $k$ andre knuder markeret med $B$.

Nedenstående graf (til venstre) viser en kvadratiske gittergraf med $g = 10$ og $k = 3$.




Afstanden mellem to knuder i grafen er antallet af kanter på en korteste sti mellem de to knuder. I ovenstående graf (til højre), er knuden $x$ den knude i grafen som ligger længst væk fra den nærmeste markerede knude. De tre nærmeste markerede knuder har afstand syv til $x$. De fede kanter angiver $k$ stier der forbinder hver af de $k$ knuder markeret med $A$ med en knude markeret med $B$.

**Spørgsmål a:** Beskriv en algoritme der finder en knude der har størst mulig afstand til den nærmeste markerede knude. Angiv algoritmens udførselstid. □

**Spørgsmål b:** Beskriv en algoritme der afgør om der findes $k$ ***kantdisjunkte*** stier, således at stierne starter i de $k$ knuder markeret med $A$ og slutter i de $k$ knuder markeret med $B$. Alle stierne skal starte i forskellige knuder og slutte i forskellige knuder. Angiv algoritmens udførselstid. (Vink: find en maksimal strømning i et passende netværk). □

**Spørgsmål c:** Beskriv en algoritme der afgør om der findes $k$ ***knude***disjunkte stier, således at stierne starter i de $k$ forskellige knude markeret med $A$ og slutter i de $k$ forskellige knuder markeret med $B$. Alle stierne skal starte i forskellige knuder og slutte i forskellige knuder. Angiv algoritmens udførselstid. □

(Opgavesættet fortsætter)

**Opgave 4** (20%)

Lad $S = x_1, \ldots, x_n$ være en sekvens af længde $n$. I denne opgave ønsker vi at finde en længste delsekvens af $S$ som er et palindrom. Et palindrom er en sekvens som er lig med sig selv læst bagfra, f.eks. `ÆRTETRÆ`. Som eksempel har `SAMHITA` delsekvensen `AHA` som en længste delsekvens der er et palindrom.

Vi lader $LP(i,j)$ betegne *længden af en længste delsekvens* af $x_i, \ldots, x_j$, der er et palindrom.

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

$$LP(i,j) = \begin{cases} 0 & \text{hvis} \quad i > j \\ 1 & \text{hvis} \quad i = j \\ 2 + LP(i+1, j-1) & \text{hvis} \quad i < j \wedge x_i = x_j \\ \max\{LP(i+1,j), LP(i,j-1)\} & \text{hvis} \quad i < j \wedge x_i \neq x_j \end{cases}$$

**Spørgsmål a:** Udfyld nedstående tabel for $LP(i,j)$ når $S =$ `ABZLA`.

| $LP(i,j)$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

□

**Spørgsmål b:** Angiv en algoritme baseret på dynamisk programmering, der givet en sekvens $S$ af længde $n$, finder længden af en længste delsekvens der er et palindrom. Angiv algoritmens udførselstid. □

**Spørgsmål c:** Udvid algoritmen fra spørgsmål b) til, givet en sekvens $S$ af længde $n$, at rapportere en længste delsekvens af $S$ der er et palindrom. Angiv algoritmens udførselstid. □

(Opgavesættet fortsætter)

**Opgave 5** (20%)

I det følgende betragter vi $k$ strenge $S_1, \ldots, S_k$ over et alfabet med $O(1)$ tegn. Hver streng har længde $n$.

Vi ønsker for hver streng $S_i$ at finde en delstreng $U_i$ således at $U_i$ ikke forekommer som delstreng i $S_j$ for $j \neq i$.

For eksempel hvis $S_1 = \texttt{ABABBACABC}$ og $S_2 = \texttt{ACCBABAABB}$, så er $U_1 = \texttt{BC}$ og $U_2 = \texttt{CB}$ delstrenge af henholdsvis $S_1$ og $S_2$ som ikke forekommer i henholdsvis $S_2$ og $S_1$.

**Spørgsmål a:** Angiv for hver af de nedenstående strenge $S_1, \ldots, S_4$, *korteste delstrenge* $U_1, \ldots, U_4$, således at $U_i$ er en delstreng af $S_i$ og $U_i$ er ikke en delstreng af nogen $S_j$ for $j \neq i$.

$$S_1 = \texttt{ABABBACABC}$$
$$S_2 = \texttt{CAACCABCAA}$$
$$S_3 = \texttt{BBBAACBCAA}$$
$$S_4 = \texttt{CCCAACCCCC}$$

□

I det følgende kan det 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 b:** Beskriv en algoritme der givet $k$ strenge $S_1, \ldots, S_k$, finder $k$ *korteste delstrenge* $U_1, \ldots, U_k$, således at $U_i$ er en delstreng af $S_i$ og $U_i$ er ikke en delstreng af nogen $S_j$ for $j \neq i$. Hver streng $S_i$ har længde $n$ og er fra et alfabet med $O(1)$ tegn. Angiv algoritmens udførselstid. □