DATALOGISK INSTITUT, AARHUS UNIVERSITET

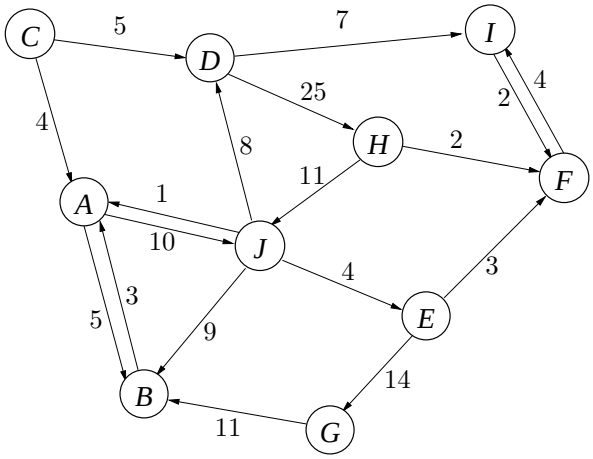
Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 8 (otte)
Eksamensdag: Fredag den 26. juni 2009, kl. 9.00-13.00
Eksamenslokale: Skøjtehallen, Gøteborg Allé 9, Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger, notater, lommeregner). Computer må ikke medbringes.
Materiale der udleveres til eksaminanden:

OPGAVETEKSTEN
BEGYNDER
PÅ NÆSTE SIDE

—oOo—

Opgave 1 (25%)

Betragt nedenstående orienterede graf med ikke-negative vægte på kanterne. Det antages at grafen er givet ved incidenslister hvor incidenslisterne er sorteret alfabetisk.



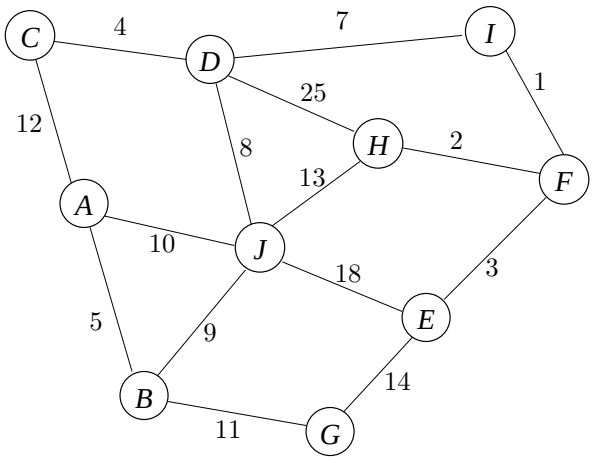
Spørgsmål a: Angiv et BFS træ for ovenstående graf, når BFS gennemløbet starter i knuden A. Angiv kanterne i BFS træet og BFS numrene for knuderne. □

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 kortest 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. □

Betragt nu følgende uorienterede graf med vægte på kanterne.

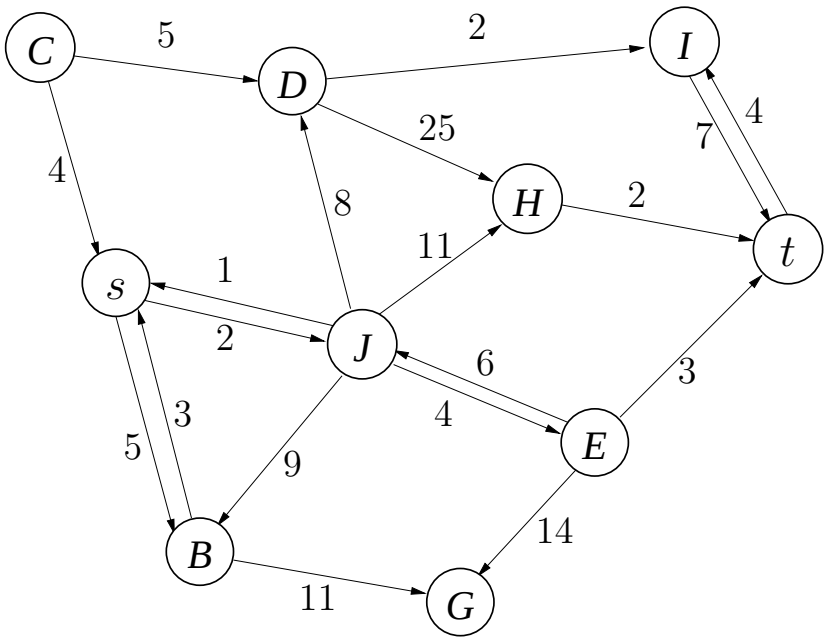


Spørgsmål e: Angiv kanterne i et minimum udspændende træ for ovenstående graf. □

(Opgavesættet fortsætter)

Opgave 2 (15%)

Betragt nedenstående netværk med de angivne kapaciteter på kanterne.



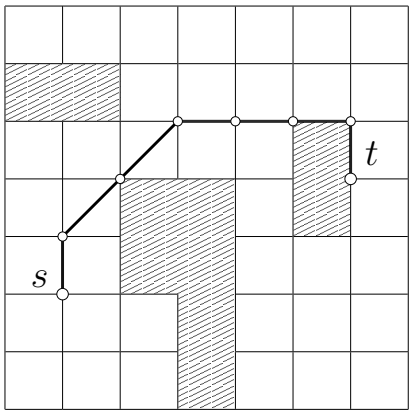
Spørgsmål a: Angiv en maksimal strømning fra s til t i netværket (angiv for hver kant strømningen langs kanten), angiv værdien af en maksimal strømning, og angiv et snit (dvs. opdeling af knuderne i to disjunkte mængder) hvor kapaciteten af snittet er lig en maksimal strømning. □

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. □

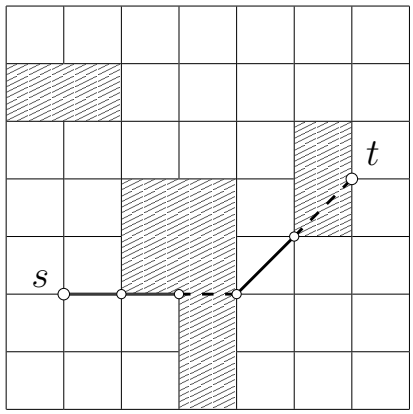
Opgave 3 (20%)

I denne opgave betragter vi et gitter af $n \times n$ celler, hvor cellerne kan være blokerede eller fri. Nedenstående eksempel er et gitter med 7×7 celler hvor skraverede celler angiver blokerede celler. Vi ønsker at finde den hurtigste sti fra et punkt s til et andet punkt t, hvor vi antager at s og t er gitterpunkter. Vi antager at input er et to-dimensionalt array som angiver for hver celle om cellen er blokeret eller fri.

Vi antager først at vi i hvert skridt går fra et gitterpunkt til et andet gitterpunkt, hvor man kan gå langs præcis en side eller en diagonal i en fri celle. Nedenstående eksempel til venstre viser en hurtigste sti fra s til t med 7 skridt. Stien i eksemplet til højre er ikke lovlig da den et sted går langs en celle-side hvor begge tilstødende celler er blokerede og et andet sted følger en diagonal i en blokeret celle.



Lovlig sti



Ulovlig sti
(ulovlige stykker af stien er stiplet)

Spørgsmål a: Beskriv en algoritme der finder en hurtigste sti fra s til t når man i hvert skridt kan gå langs præcis en side eller en diagonal i en fri celle. Angiv algoritmens udførselstid. □

Vi betragter nu en variation af problemet hvor man i hvert skridt anvender en hastighedsvektor $v_i = p_{i+1} - p_i$, hvor $p_0, p_1, \ldots, p_k$ er stien af gitterpunkter man besøger og $p_0 = s$ og $p_k = t$.

For en lovlig sti kræver vi at:

(a) alle v_i har heltallige koordinater (garanterer at hvert skridt ender på et gitterpunkt)

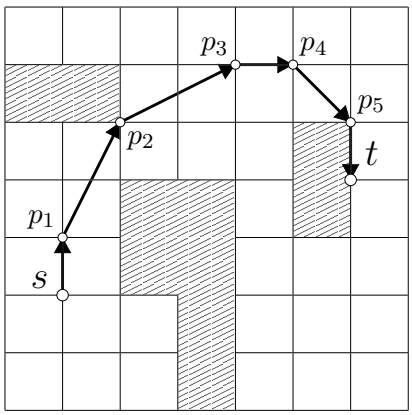
(b) differensen $v_{i+1} - v_i$ mellem to hastighedsvektorer har koordinater i $\{-1, 0, 1\}$ (garanterer begrænset acceleration)

(c) koordinaterne på den første og sidste hastighedsvektor tilhører $\{-1, 0, 1\}$ (garanterer at vi kan starte og slutte med hastighed 0)

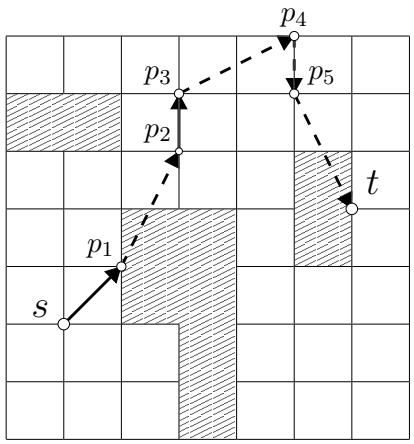
(d) den sammenhængende sti fra s til t, hvor p_i er forbundet til p_{i+1} med en lige linie, går kun igennem det indre eller siderne på frie celler.

I nedenstående eksempel til venstre er de 6 hastighedsvektorer (0,1), (1,2), (2,1), (1,0), (1,-1) og (0,-1), og i eksemplet til højre er de 6 hastighedsvektorer (1,1), (1,2), (0,1), (2,1), (0,-1), (1,-2).

Eksemplet til højre opfylder ikke kravene til en lovlig sti da følgende ikke er opfyldt: (b) gælder ikke da hastighedsvektoren ændres fra (0,1) til (2,1) i p_3, og tilsvarende i p_4 ændres hastighedsvektoren fra (2,1) til (0,-1); (c) gælder ikke for den sidste hastighedsvektor, som er (1,-2); (d) gælder ikke mellem p_1 og p_2 og mellem p_5 og t hvor stien går igennem det indre af to blokerede celler.



Lovlig sti



Ulovlig sti
(ulovlige stykker af stien er stiplet)

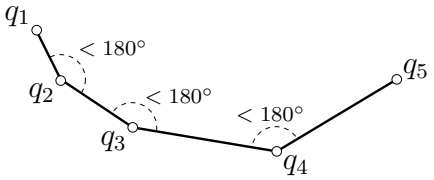
Spørgsmål b: Beskriv en algoritme der finder en hurtigste sti fra s til t når hvert skridt har sin egen hastighedsvektor. Angiv algoritmens udførselstid. Hint: Lav en graf med en knude for hvert par af et gitterpunkt og en hastighedsvektor. □

(Opgavesættet fortsætter)

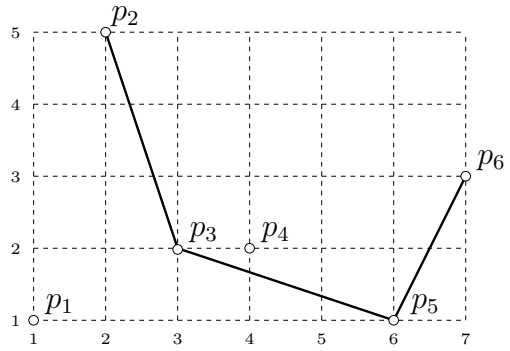
Opgave 4 (20%)

Lad $S = \{p_1, \ldots, p_n\}$ være en mængde af $n \geq 2$ punkter i planen, hvor $p_i = (x_i, y_i)$ og x_i og y_i er heltal. Det antages at $x_i < x_{i+1}$ for alle $1 \leq i < n$, dvs. punkterne har forskellige x-koordinater og punkterne er sorteret efter x-koordinaterne.

En sekvens af punkter $Q = q_1, \ldots, q_k$ betegnes som en *konveks kæde* hvis punkterne er sorteret efter x-koordinaterne, x-koordinaterne er forskellige, og for alle $1 \leq i \leq k-2$ udgør punkterne q_i, q_{i+1}, q_{i+2} en *venstre drejning*, dvs. vinklen mellem linierne $q_{i+1}q_i$ og $q_{i+1}q_{i+2}$ er mindre end 180°.



Hvis $Q \subseteq S$ siger vi at Q er en konveks kæde i S. I nedenstående eksempel er $Q = \{p_2, p_3, p_5, p_6\}$ en konveks kæde i $S = \{p_1, p_2, p_3, p_4, p_5, p_6\}$.



For $1 \leq i < j \leq n$ lader vi $CC(i,j)$ betegne *størrelsen af en største konveks kæde i S hvor p_i og p_j er de to sidste punkter i kæden.*

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

$$CC(i,j) = \begin{cases} \max\{CC(k,i)+1 \mid 1 \leq k < i \wedge \text{LeftTurn}(p_k, p_i, p_j)\} \\ \qquad \text{hvis} \quad i > 1 \wedge \text{der findes } k' \text{ hvor } 1 \leq k' < i \wedge \text{LeftTurn}(p_{k'}, p_i, p_j) \\ 2 \qquad \text{ellers} \end{cases}$$

Her angiver $\text{LeftTurn}(p_k, p_i, p_j)$ testet for om tre punkter udgør en venstre drejning, som kan testes ved følgende udtryk:

$$x_i y_j - x_j y_i - x_k y_j + x_j y_k + x_k y_i - x_i y_k > 0$$

(Opgavesættet fortsætter)

Spørgsmål a: Udfyld for ovenstående punktmængde indgangene i nedenstående tabel for $CC(i, j)$ for $i < j$.

$CC(i,j)$	2	3	4	5	6
1					
2					
3					
4					
5					

□

Spørgsmål b: Angiv en algoritme baseret på dynamisk programmering, der givet en mængde S med n punkter, finder størrelsen af en største konveks kæde i S. Angiv algoritmens udførselstid. □

Spørgsmål c: Udvid algoritmen fra spørgsmål b) til, givet en mængde S med n punkter, at rapportere punkterne i en størst mulig konveks kæde i S. Angiv algoritmens udførselstid. □

(Opgavesættet fortsætter)

Opgave 5 (20%)

For en streng $U = x_1x_2 \cdots x_{k-1}x_k$ betegner $\overleftarrow{U} = x_kx_{k-1} \cdots x_2x_1$ strengen U læst bagfra. F.eks. for $U = \texttt{ABABBC}$ så er $\overleftarrow{U} = \texttt{CBBABA}$.

For strengen $S = \texttt{ACBABCBAAABC}$ forekommer stregen $U = \texttt{ABC}$ på position 4 og 10 og $\overleftarrow{U} = \texttt{CBA}$ på position 2 og 6.

$$S = \overleftarrow{\texttt{A}}\ \overleftarrow{\texttt{C}}\ \overleftarrow{\texttt{B}}\ \overrightarrow{\texttt{A}}\ \overleftarrow{\texttt{B}}\ \overrightarrow{\texttt{C}}\ \overleftarrow{\texttt{B}}\ \texttt{A}\ \texttt{A}\ \overrightarrow{\texttt{A}}\ \overrightarrow{\texttt{B}}\ \overrightarrow{\texttt{C}}$$
$$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12$$

Spørgsmål a: Angiv for strengen

$$S = \texttt{ABBACADABA}$$

alle delstrenge U hvor både U og $\overleftarrow{U}$ forekommer mindst en gang i S og $|U| \geq 2$. For hver delstreng U angiv U samt de positioner hvor U og $\overleftarrow{U}$ forekommer.

U	Forekomster U	Forekomster $\overleftarrow{U}$

□

I det følgende kan det antages at et suffiks-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 en streng S af længde n fra et alfabet med $O(1)$ tegn, finder en længste delstreng U hvor både U og $\overleftarrow{U}$ forekommer i S. Angiv algoritmens udførselstid. □

Spørgsmål c: Beskriv en algoritme der givet en streng S af længde n fra et alfabet med $O(1)$ tegn, finder en delstreng U hvor $|U| \geq 2$ og antallet af forekomster af U i S er lig antallet af forekomster af $\overleftarrow{U}$ i S, eller rapporterer at en sådan delstreng findes ikke. Angiv algoritmens udførselstid. □