DATALOGISK INSTITUT, AARHUS UNIVERSITET

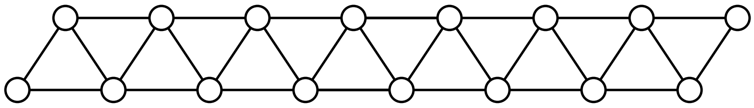
Det Naturvidenskabelige Fakultet
EKSAMEN
Grundkurser i Datalogi
Algoritmer og Datastrukturer 2 (2003-ordning)
Antal sider i opgavesættet (incl. forsiden): 7 (syv)
Eksamensdag: Mandag den 20. juni 2005, kl. 9.00-13.00
Eksamenslokale: Skøjtehallen, Gøteborg Allé 9, Århus N
Tilladte medbragte hjælpemidler: Alle sædvanlige hjælpemidler (lærebøger og notater)
Materiale der udleveres til eksaminanden:

OPGAVETEKSTEN
BEGYNDER
PÅ NÆSTE SIDE

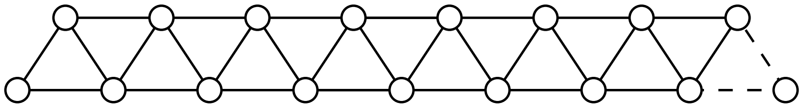
—oOo—

Opgave 1 (25%)

I denne opgave betragtes *k-trekant-kæde* grafer, som består af k trekanter i en kæde, hvor en trekant deler en kant med den forrige og den næste trekant. Nedenstående viser en 14-trekant-kæde graf.



Bemærk at en $(k+1)$-trekant-kæde graf fremkommer ved at tilføje en trekant til en k-trekant-kæde graf, j.v.f. nedenstående 15-trekant-kæde graf.



Spørgsmål a: Angiv antal knuder og antal kanter i en k-trekant-kæde graf som funktion af antal trekanter k. Angiv udførselstiden for Kruskal's algoritme for en vægtet k-trekant-kæde graf. Udførselstiden skal angives som funktion af k. □

Lad $G = (V, E)$ være en vægtet uorienteret graf, og lad $K \subseteq E$ være en kantmængde, som indeholder et letteste udspændende træ for G. Hvis K indeholder en cykel C og hvis e er en kant i C af størst vægt, da vil $K \setminus \{e\}$ også indeholde et letteste udspændende træ for G.

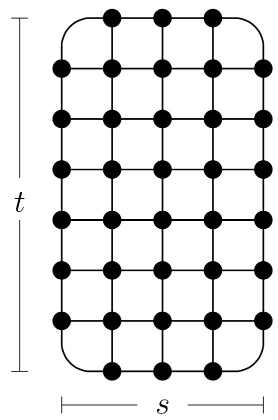
Spørgsmål b: Beskriv en algoritme med udførselstid $O(n)$, der finder et letteste udspændende træ for en vægtet k-trekant-kæde graf med n knuder. Argumenter for algoritmens udførselstid. □

Spørgsmål c: Angiv en algoritme med udførselstid $O(n)$, der finder den *letteste simple cykel* i en k-trekant-kæde graf hvor alle kanter har positive vægte. Argumenter for algoritmens udførselstid. □

(Opgavesættet fortsætter)

Opgave 2 (25%)

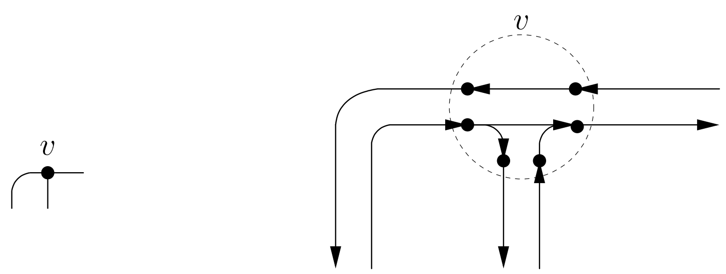
I denne opgave betragtes *Manhattan*-grafer. En Manhattan-graf er en uorienteret graf der består af t rækker med s knuder, men hvor de fire hjørner mangler, d.v.s. den første og den sidste række har kun $s-2$ knuder. Knuderne er forbundet som vist på nedenstående graf der viser en Manhattan-graf for $s = 5$ og $t = 8$. Bemærk at alle knuder har grad tre eller fire, og at indgående kanter til en knude kommer enten fra nord, syd, øst eller vest.



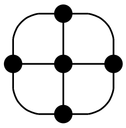
Spørgsmål a: Angiv antallet af kanter og knuder i en Manhattan-graf som funktion af s og t. Angiv udførselstiden for Dijkstra's algoritme for at beregne den korteste vej fra en knude u til en knude v i en vægtet Manhattan-graf, hvor alle kanter har positive vægte. Udførselstiden skal angives som funktion af s og t. □

Betragt en Manhattan-graf. Der antages at kanter repræsenterer veje og man for alle veje kan køre i begge retninger, men at når man kommer ad en vej til en knude *kan man ikke dreje til venstre*, d.v.s. i en knude kan man kun følge vejen lige ud eller dreje til højre.

En uorienteret Manhattan-graf kan transformeres til en orienteret graf ved at erstatte en knude af grad 3 eller 4 med henholdsvis 6 eller 8 knuder i en orienteret graf, således at alle stier i den orienterede graf svarer til veje der er lovlige at køre i Manhattan-grafen. Nedenstående viser hvordan en knude v af grad 3 bliver til 6 knuder og de incidente kanter hertil.

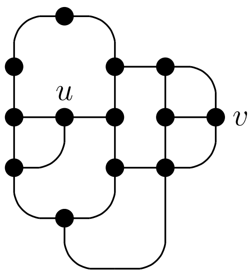


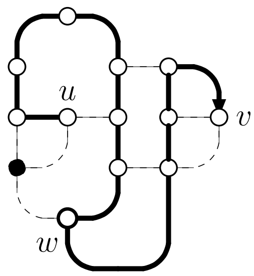
Spørgsmål b: For nedenstående Manhattan-graf, angiv den tilsvarende orienterede graf hvor stier svarer til lovlige veje at køre i Manhattan-grafen.



□

I de næste to spørgsmål betragtes orienterede grafer hvor alle kanterne incidente til en knude kommer fra nord, syd, øst eller vest. Der kommer højst én kant fra hver af de fire retninger. Nedenstående graf til venstre viser et eksempel på sådan en graf. Bemærk at der *ikke* findes en vej fra u til v hvor man aldrig drejer til venstre, uanset hvilken kant incident til u man starter med. Til højre ses en sti fra u til v der foretager ét venstre sving, nemlig i w.





Spørgsmål c: Beskriv en algoritme, der givet en uorienteret graf hvor alle kanterne incidente til en knude kommer fra nord, syd, øst eller vest, afgør om der findes en vej fra en knude u til en knude v som *ikke indeholder nogen venstre sving* (man må selv bestemme hvilken kant incident med u man starter med). Angiv algoritmens udførselstid som funktion af antal knuder n. □

Spørgsmål d: Beskriv en algoritme, der givet en uorienteret graf hvor alle kanterne incidente til en knude kommer fra nord, syd, øst eller vest, finder en vej fra en knude u til en knude v som indeholder *færrest mulige venstre sving* (man må selv bestemme hvilken kant incident med u man starte med). Angiv algoritmens udførselstid som funktion af antal knuder n. □

Opgave 3 (25%)

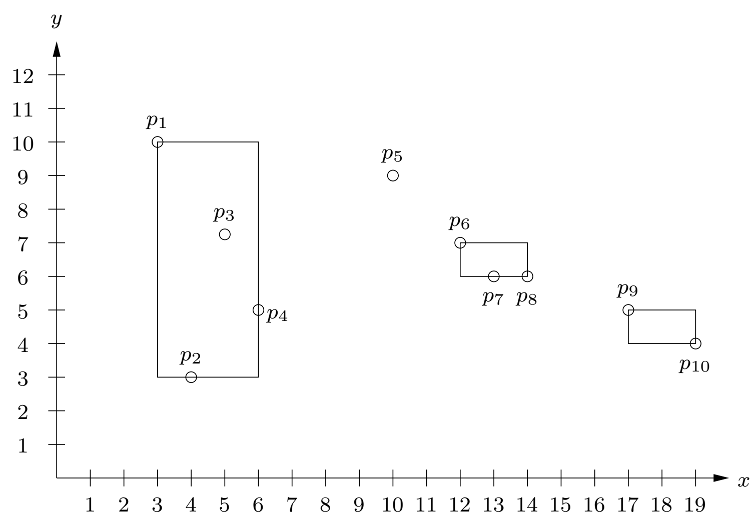
I denne opgave betragtes n punkter

$$p_1 = (x_1, y_1),\ p_2 = (x_2, y_2), \ldots,\ p_n = (x_n, y_n)$$

hvorom der gælder $x_1 < x_2 < x_3 < \cdots < x_n$.

For et givet k, ønsker vi at finde k rektangler der tilsammen indeholder alle punkterne og således at det samlede *areal* af rektanglerne er mindst muligt. Siderne på de k rektangler skal være parallelle med x- og y-akserne, og der skal gælde at rektanglerne ikke overlapper m.h.t. x-værdierne, d.v.s. rektanglerne kan sorteres fra venstre mod højre.

Nedenstående viser et eksempel hvor 10 punkter er indeholdt i 4 rektangler, med henholdsvis punkterne $\{p_1, p_2, p_3, p_4\}$, $\{p_5\}$, $\{p_6, p_7, p_8\}$, og $\{p_9, p_{10}\}$. Bemærk at rektanglet indeholdende p_5 kun består af punktet p_5.



Det samlede areal af de fire rektangler er

$$(6-3) \times (10-3) + (10-10) \times (9-9) + (14-12) \times (7-6) + (19-17) \times (5-4) = 25$$

Lad $B(i, j)$ betegne arealet af det minimale rektangel som indeholder $p_i, p_{i+1}, \ldots, p_{j-1}, p_j$ og som kan beregnes ved formlen

$$B(i, j) = (x_j - x_i) \times \left(\max_{k\,:\,i \le k \le j} y_k - \min_{k\,:\,i \le k \le j} y_k \right)$$

For ovenstående punktmængde er f.eks. $B(6, 8) = (14 - 12) \times (7 - 6) = 2$.

Spørgsmål a: Angiv en algoritme der beregner $B(i, j)$ for alle par (i, j) hvor $i \le j$. Angiv algoritmens udførselstid. $\square$

For $s \geq 1$ og $t \geq 1$ definerer vi nu:

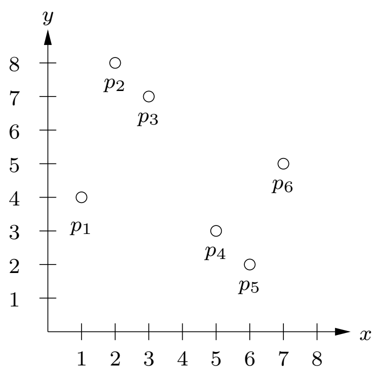
$$A(s,t) = \text{det minimale areal af op til } s \text{ rektangler}$$
$$\text{der indeholder punkterne } p_1, p_2, \ldots, p_t.$$

Bemærk at $A(k,n)$ er det minimale areal af op til k rektangler der indeholder alle n punkter.

Det påstås at $A(s,t)$ opfylder følgende rekursionsformel:

$$A(s,t) = \begin{cases} B(1,t) & \text{hvis} \quad s = 1 \ \vee \ t = 1 \\ \\ \min\limits_{k\,:\,2 \leq k \leq t} \left(A(s-1,k-1) + B(k,t)\right) & \text{hvis} \quad t \geq 2 \ \wedge \ s \geq 2 \end{cases}$$

Spørgsmål b: Udfyld nedenstående tabel for $A(s,t)$, $1 \leq s \leq 3$ og $1 \leq t \leq 6$, for punktmængde $\{(1,4),(2,8),(3,7),(5,3),(6,2),(7,5)\}$.



$s \backslash t$	1	2	3	4	5	6
1						
2						
3						

□

Spørgsmål c: Angiv en algoritme med udførselstid $O(n^3)$ baseret på dynamisk programmering, der givet n punkter og et heltal k, beregner $A(k,n)$, d.v.s. det minimale areal af k rektangler der indeholder alle n punkter. Argumenter for algoritmens udførselstid. □

Opgave 4 (25%)

For en streng $S = S[1]S[2]\cdots S[n]$ er *i-rotationen* af S strengen

$$\text{rotate}_i(S) = S[i+1..|S|]S[1..i]$$

F.eks. har strengen *abcdefgh* 3-rotationen *defghabc*.

En streng S er *en rotation af sig selv* hvis der findes i, $1 \leq i < n$, således at $S = \text{rotate}_i(S)$. F.eks. er $S = abcdabcd$ en rotation af sig selv da $S = \text{rotate}_4(S)$.

Spørgsmål a: Beskriv hvordan man kan anvende en mønster genkendelses algoritme, f.eks. Knuth-Morris-Pratt, til at afgøre om en streng S af længde n er en rotation af sig selv. Angiv algoritmens udførselstid. Hint: Betragt strengen SS. □

Betragt nu en streng S af længde n og et mønster P af længde m, begge over et alfabet af størrelse $O(1)$.

For mønsteret $P =$*abcde* forekommer $\text{rotate}_2(P)$ i strengen $S =$*abdbcabdebva*<u>*cdeab*</u>*abcd* startende ved position 13.

Spørgsmål b: Angiv alle forekomster af mønsteret $P =$*aba* eller en rotation af P i strengen $S =$ *abaaaba*. □

Spørgsmål c: Angiv suffix-træet for strengen $PPS\$$ hvor P og S er de i spørgsmål b) anvendte strenge, d.v.s. suffix-træet for strengen *abaabaabaaaba\$*. Symbolet \$ er et symbol der ikke er med i alfabetet. Børnene til en knude skal komme i alfabetisk rækkefølge, fra venstre mod højre, hvor det antages at \$ er det sidste tegn i alfabetet. □

I det følgende kan antages uden bevis at et suffix-træ for en streng af længde n over et alfabet af størrelse $O(1)$ kan konstrueres i tid $O(n)$.

Spørgsmål d: Beskriv en algoritme der afgør om en rotation af mønsteret P forekommer i S. Angiv algoritmens udførselstid. Hint: Betragt suffix-træet for strengen $PPS\$$ af længde $2m + n$. □