Skriftlig Eksamen
Algoritmer og Datastrukturer 2 (2003-ordning)

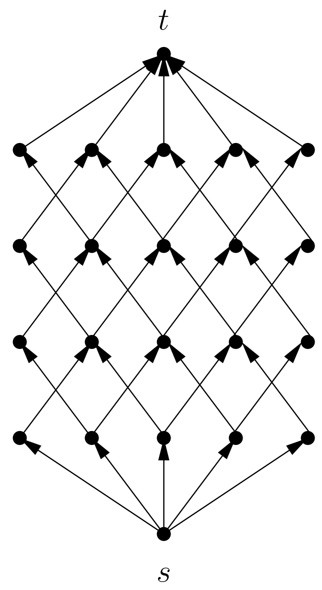
Datalogisk Institut
Aarhus Universitet

Fredag den 28. maj 2004, kl. 9.00–13.00

Opgave 1 (20%)

En (r, k) *kryds-graf* er en orienteret graf med r rækker af k knuder, hvor knude j i række i, for $1 \leq i < r$, har en kant til knude $j-1$ i række $i+1$, for $1 < j \leq k$, og en kant til knude $j+1$ i række $i+1$, for $1 \leq j < k$. Desuden findes en knude s, der har kanter til alle knuder i den første (nederste) række, og en knude t, der har kanter fra alle knuder i den sidste (øverste) række. Lad m betegne antal kanter og n antal knuder i en (r, k) kryds-graf

Nedenstående viser en $(4, 5)$ kryds-graf.



Spørgsmål a: Angiv antal kanter m og antal knuder n i en (r, k) kryds-graf som funktion af r og k. Antag alle kanter har en ikke negativ vægt. Angiv udførselstiden af Dijkstra's algoritme til at finde de korteste afstande fra s til alle de øvrige knuder som funktion af r og k. $\square$

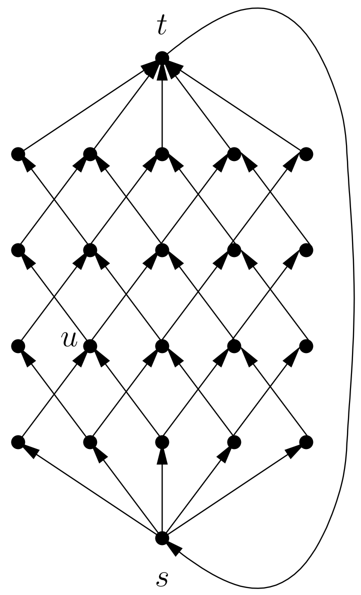
Spørgsmål b: Angiv en $(1, 2)$ kryds-graf med positive og negative vægtede kanter, hvor Dijkstra's algoritme beregner en forkert afstand fra s til t. Angiv for hver knude i eksemplet de korrekte afstande fra s samt de afstande Dijkstra's algoritme beregner. $\square$

Spørgsmål c: Angiv en algoritme med udførselstid $O(n)$ til at beregne de korteste afstande fra s til alle de øvrige knuder i en (r, k) kryds-graf med positive og negative vægtede kanter. Argumenter for algoritmens udførselstid. $\square$

Opgave 2 (20%)

En *cyklisk* (r, k) *kryds-graf* er en orienteret graf med r rækker af k knuder, hvor knude j i række i, for $1 \leq i < r$, har en kant til knude $j-1$ i række $i+1$, for $1 < j \leq k$, og en kant til knude $j+1$ i række $i+1$, for $1 \leq j < k$. Desuden findes en knude s, der har kanter til alle knuder i den første (nederste) række, en knude t, der har kanter fra alle knuder i den sidste (øverste) række, og en kant fra t til s. Lad m betegne antal kanter og n antal knuder i en cyklisk (r, k) kryds-graf

Nedenstående viser en cyklisk $(4, 5)$ kryds-graf.



I denne opgave betragtes problemet at finde de korteste afstande fra en vilkårlig knude u til alle de øvrige knuder i en cyklisk (r, k) kryds-graf, hvor alle kanter har vægte der kan være både *positive og negative*. Der antages at der ikke findes negative cykler.

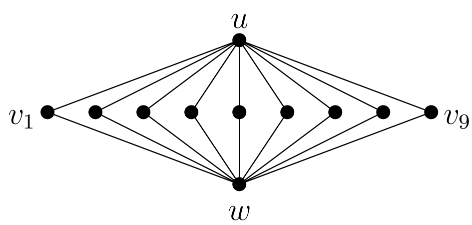
Spørgsmål a: Angiv som funktion af r og k udførselstiden af Bellman-Ford's algoritme for at finde de korteste afstande fra en knude u til alle de øvrige knuder i en cyklisk (r, k) kryds-graf. □

Spørgsmål b: Angiv en algoritme med udførselstid $O(n)$, der finder de korteste afstande fra en knude u til alle de øvrige knuder i en cyklisk (r, k) kryds-graf. Argumenter for algoritmens udførselstid. □

Spørgsmål c: Angiv en algoritme med udførselstid $O(n^2)$, der finder de korteste afstande mellem alle par af knuder i en cyklisk (r, k) kryds-graf. Argumenter for algoritmens udførselstid. □

Opgave 3 (10%)

En n-*dobbeltstjerne-graf* er en uorienteret graf med to knuder u og w og n knuder $v_1, \ldots, v_n$. For hver af knuderne $v_1, \ldots, v_n$ er der kanter til u og w. Nedenstående er en 9-dobbeltstjerne-graf



Man kan i det følgende bruge nedenstående sætning om minimum udspændende træer uden bevis:

> *Lad $G = (V, E)$ være en vægtet uorienteret graf, og lad $K \subseteq E$ være en kantmængde, som indeholder et minimum 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 minimum udspændende træ for G.*

Spørgsmål a: Beskriv en algoritme med udførselstid $O(n)$, der finder et minimum udspændende træ for en vægtet n-dobbeltstjerne-graf. Argumenter for algoritmens udførselstid. Hint: Betragt cykler (u, v_i, w, v_j, u) hvor $i \neq j$. □

Opgave 4 (25%)

Lad $S = \{x_1, \ldots, x_n\}$ være en mængde af n positive heltal og N et positivt heltal, hvor $N \geq x_1 + \cdots + x_n$. Opgaven går ud på at afgøre om der findes positive heltal $d_1, \ldots, d_n$ således at $N = {x_1}^{d_1} + \cdots + {x_n}^{d_n}$.

For mængden $S = \{1, 2, 3, 5, 7\}$ og $N = 740$ findes løsningen

$$740 = 1^1 + 2^7 + 3^5 + 5^2 + 7^3 \ .$$

Lad $B(k, s)$ være sand hvis og kun hvis der findes positive $d_1, \ldots, d_k$ så $s = {x_1}^{d_1} + \cdots + {x_k}^{d_k}$. Bemærk at $B(n, N)$ er svaret på det oprindelige problem.

Følgende er en rekursiv opskrivning af $B(k, s)$, hvor s og k er heltal og $0 \leq k \leq n$ og $0 \leq s \leq N$.

$$B(k,s) = \begin{cases} \textbf{sand} & \text{hvis} \quad k = 0 \wedge s = 0 \\ \textbf{falsk} & \text{hvis} \quad k = 0 \wedge s > 0 \\ \textbf{sand} & \text{hvis} \quad k > 0 \wedge (\text{der findes } i \geq 1 \text{ så } {x_k}^i \leq s \wedge B(k-1, s - {x_k}^i)) \\ \textbf{falsk} & \text{ellers} \end{cases}$$

Spørgsmål a: Udfyld nedenstående tabel for $B(k, s)$ for mængden $S = \{2, 3, 4\}$ og $N = 21$ (angiv blot indgange der har værdien sand).

$k \setminus s$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0																						
1																						
2																						
3																						

□

Spørgsmål b: Angiv en algoritme baseret på dynamisk programmering, der afgør om $B(n, N)$ er sand. Angiv algoritmens udførselstid og argumenter for denne. □

Spørgsmål c: Udvid algoritmen fra spørgsmål (b) til i samme udførselstid at udskrive en løsning $d_1, \ldots, d_n$, såfremt der findes en løsning. Angiv algoritmens udførselstid og argumenter for denne. □

Opgave 5 (25%)

Lad S_1 og S_2 være to strenge med tilsammen n tegn over alfabetet $\{a, b, c\}$, dvs. $n = |S_1| + |S_2|$. I denne opgave betragtes problemet at finde den længste (sammenhængende) fælles delstreng for S_1 og S_2.

Lad $S = S_1\#S_2\$$ være en ny streng over alfabetet $\{a, b, c, \#, \$\}$, som består af de to strenge sat efter hinanden, hvor S_1 og S_2 er adskildt af et nyt symbol # og S_2 er efterfulgt af et nyt symbol $.

Eksempel: For strengene $S_1 = a\,a\,b\,a\,c$ og $S_2 = a\,c\,a\,b\,a\,b$ er den længste fælles delstreng $a\,b\,a$, og $S = a\,a\,b\,a\,c\,\#\,a\,c\,a\,b\,a\,b\,\$$.

Spørgsmål a: Opskriv alle suffixerne af strengen S fra ovenstående eksempel i leksikografisk rækkefølge, hvor alfabetets ordning er $a < b < c < \# < \$$. □

Spørgsmål b: Angiv suffix-træet for strengen S fra ovenstående eksempel, og marker knuden svarende til den længste fælles delstreng af S_1 og S_2. □

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 c: Beskriv en algoritme der givet to strenge med totalt n tegn over et alfabet af størrelse $O(1)$ kan finde den længste fælles delstreng i tid $O(n)$. Argumenter for algoritmens udførselstid. Hint: Marker knuderne i suffix-træet med information om der findes delstrenge i S_1 og S_2 der svarer til stierne fra roden ned til knuderne. □