Skriftlig Eksamen
Algoritmer og Datastrukturer (dADS)

Datalogisk Institut
Aarhus Universitet

Onsdag den 13. august 2003, kl. 9.00–13.00

Opgave 1 (25%)

Lad $A = A[1] \cdots A[n]$ være et array af heltal. Længden af det længste sammenhængende delarray af A hvor alle indgange er identiske betegnes LID(A). Længden af det længste suffix af A hvor alle indgange er identiske betegnes LIDS(A).

Eksempel:

$$\mathrm{LID}(\texttt{1,2,3,3,2,2,}\underline{\texttt{1,1,1,1,1}}\texttt{,2,2,4,4,2,2,2,2}) = 5$$
$$\mathrm{LIDS}(\texttt{1,2,3,3,2,2,1,1,1,1,1,2,2,4,4,}\underline{\texttt{2,2,2,2}}) = 4$$

Betragt følgende algoritme:

Algoritme : LID

Input:	A : Array af længde n, hvor $n \geq 1$
Output:	m : LID(A)
Metode:	$r \leftarrow 1$;
	$m \leftarrow 1$;
	$i \leftarrow 1$;
	$\{ I \}$ **while** $i < n$ **do**
	$\ll$ *Update* $\gg$;
	$i \leftarrow i + 1$;

hvor I er invarianten

$$(r = \mathrm{LIDS}(A[1] \cdots A[i])) \ \wedge \ (m = \mathrm{LID}(A[1] \cdots A[i])) \ \wedge \ (i \leq n)$$

Spørgsmål a: Angiv hvilke bevisbyrder der skal eftervises i et gyldighedsbevis for algoritmen. □

Spørgsmål b: Angiv konkret kode for $\ll$ *Update* $\gg$, således at bevisbyrderne fra spørgsmål **a** kan eftervises. □

Spørgsmål c: Eftervis bevisbyrderne fra spørgsmål **a**, og argumenter for at algoritmen er korrekt. □

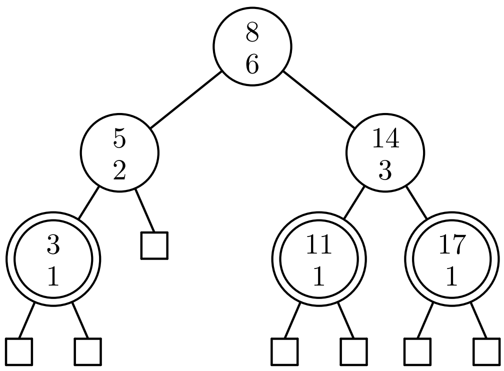
Opgave 2 (25%)

Lad $x_1, x_2, \cdots, x_n$ være n forskellige heltallige elementer. Opgaven beskæftiger sig med at understøtte rang-baserede operationer.

Operationen $\mathsf{rank}(y)$ skal returnere antallet af elementer blandt $x_1, \ldots, x_n$ der er mindre end eller lig med y, dvs. $|\{i \mid x_i \leq y\}|$.

Operationen $\mathsf{atRank}(r)$ skal, for $1 \leq r \leq n$, returnere det $r$te mindste element blandt $x_1, \ldots, x_n$, dvs. det x_j hvor $\mathsf{rank}(x_j) = r$.

I det følgende betragtes en udvidelse af rød-sorte søgetræer til opbevaring af elementerne. Lad $v.x$ være elementet gemt i knuden v. Hver knude v udvides til også at gemme $v.s$, der angiver antallet af elementer gemt i undertræet med rod i v. Nedenstående er et udvidet rød-sort søgetræ for elementerne 3,5,8,11,14,17. I knuderne er øverst angivet $v.x$ og nederst $v.s$. Røde knuder er markeret med dobbelt-cirkler. I det givne eksempel er $\mathsf{rank}(10) = 3$ og $\mathsf{atRank}(5) = 14$.



Spørgsmål a: Beskriv hvordan $\mathsf{rank}(y)$ og $\mathsf{atRank}(r)$ kan udføres i tid $O(\log n)$. Argumenter for algoritmernes udførselstid og korrekthed. □

Spørgsmål b: Beskriv hvordan en operation $\mathsf{Count}(y, z)$, der returner antallet af elementer i intervallet $]y\,;z]$, dvs. $|\{i \mid y < x_i \leq z\}|$, kan udføres i tid $O(\log n)$. Argumenter for algoritmens udførselstid og korrekthed. □

Spørgsmål c: Beskriv hvordan et sådant udvidet rød-sort søgetræ kan vedligeholdes under indsættelse og sletning af elementer i tid $O(\log n)$. □

Opgave 3 (25%)

Betragt følgende situation: En tømrer ønsker at udmåle en bestemt længde, men har glemt sin tommestok. Til gengæld har han forskellige æsker med søm i kendte størrelser, og får den idé at udmåle længden ved at lægge søm på række efter hinanden.

I denne opgave ønsker vi at beregne, hvilke længder tømreren kan udmåle med en given beholdning af søm.

Vi antager at der er K æsker med søm, og at der i æske nummer i er n_i søm, alle med en længde på l_i centimeter. Vi ønsker at afgøre om der findes en delmængde af sømmene, hvis samlede længde er S centimeter, dvs. om der findes heltal $x_1, \ldots, x_K$, hvor $0 \leq x_i \leq n_i$, således at $\sum_{i=1}^{K} x_i \cdot l_i = S$. Vi antager at S og l_i'erne alle er positive heltal.

Betragt følgende udsagn $U(k, s)$ for $0 \leq k \leq K$ og $0 \leq s \leq S$:

$$U(k, s) : \text{Der findes heltal } x_1, \ldots, x_k, \text{ hvor } 0 \leq x_i \leq n_i, \text{ så } \sum_{i=1}^{k} x_i \cdot l_i = s\ .$$

Spørgsmål a: Lad $S = 10$, $K = 3$, $l_1 = 1$, $l_2 = 5$, $l_3 = 4$, $n_1 = 3$, $n_2 = 1$, og $n_3 = 25$. Udfyld nedenstående tabel med sandhedsværdierne for $U(k, s)$ for $0 \leq k \leq K$ og $0 \leq s \leq S$.

$k \backslash s$	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											

□

Det påstås at $U(k, s)$ opfylder følgende rekursionsformel.

$$U(k,s) = \begin{cases} \textbf{Sand} & \text{hvis} \quad k = 0 \ \wedge \ s = 0 \\ \textbf{Falsk} & \text{hvis} \quad k = 0 \ \wedge \ s > 0 \\ \textbf{Sand} & \text{hvis} \quad k > 0 \ \wedge \ \text{der findes } x_k : \\ & \qquad (0 \leq x_k \leq n_k) \ \wedge \ (x_k \cdot l_k \leq s) \ \wedge \ U(k-1, s - x_k \cdot l_k) \\ \textbf{Falsk} & \text{ellers} \end{cases}$$

Spørgsmål b: Argumenter for ovenstående påstand. □

Spørgsmål c: Angiv en algoritme, baseret på dynamisk programmering, der beregner værdien $U(K, S)$. Argumenter for algoritmens udførselstid. □

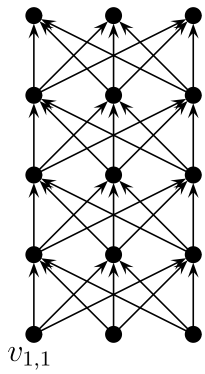
Opgave 4 (25%)

En $s \times t$-*lagdelt graf* er en orienteret graf hvor knuderne er arrangeret i s rækker hver indeholdende t knuder, hvor s og t er positive heltal. Lad $v_{i,j}$ betegne den $j$te knude i den $i$te række. En $s \times t$-lagdelt graf har følgende knuder og kanter:

$$V = \{v_{i,j} \mid 1 \leq i \leq s \wedge 1 \leq j \leq t\}$$

$$E = \{(v_{i,j}\,,\, v_{i+1,k}) \mid 1 \leq i < s \wedge 1 \leq j \leq t \wedge 1 \leq k \leq t\}$$

Nedenstående figur viser en 5×3-lagdelt graf.



Spørgsmål a: Lad n og m betegne henholdsvis antallet af knuder og kanter i en $s \times t$-lagdelt graf. Hvad er n og m som funktioner af s og t?

Vi antager nu at alle kanter har en ikke-negativ vægt. Hvad er udførselstiden som funktion af s og t for Dijkstra's algoritme til at finde korteste afstand fra $v_{1,1}$ til alle de øvrige knuder i en $s \times t$-lagdelt graf? □

I resten opgaven antages at vægtene på kanterne er reelle tal (både positive og negative).

Spørgsmål b: Beskriv hvordan man kan finde korteste afstand fra $v_{1,1}$ til alle de øvrige knuder i en $s \times t$-lagdelt graf i tid $O(st^2)$. □

Spørgsmål c: Angiv som funktion af s og t udførselstiden i en $s \times t$-lagdelt graf for den variant af Floyd-Warshalls algoritme, der finder korteste afstand mellem alle par af knuder. Anvend løsningen fra spørgsmål **b** til at beskrive en mere effektiv algoritme, og argumenter for dennes udførselstid. □

Spørgsmål d: Når kanterne kan have negative vægte, vil Dijktra's algoritme ikke nødvendigvis finde korteste afstand fra $v_{1,1}$ til alle de øvrige knuder. Angiv en vægtet 3×2-lagdelt graf hvor Dijkstra's algoritme fejler, og argumenter for at den fejler på denne graf. □