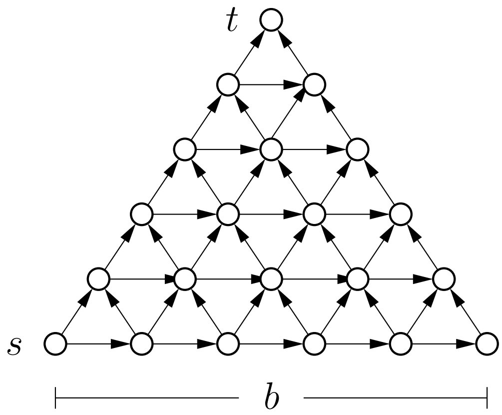
Opgave 1 (25%)

I denne opgave betragter vi *trekanter-grafer*. En trekant-graf er en orienteret graf med b rækker af knuder med henholdsvis $1, 2, 3, \ldots, b$ knuder, og hvor kanterne er som angivet i nedenstående eksempel. Nedenstående viser en trekant-graf for $b = 6$.



Spørgsmål a: Angiv antallet *kanter* m og *knuder* n i en trekant-graf som funktion af b. Angiv udførselstiden for *Dijkstra's algoritme* for at beregne den korteste afstand fra den nedre venstre knude s til alle de øvrige knuder i en positiv vægtet trekant-graf. Udførselstiden skal angives som funktion af b. □

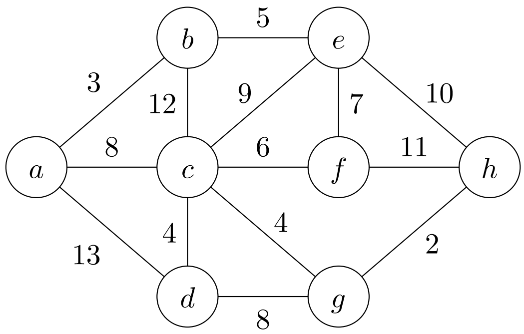
Spørgsmål b: Angiv en trekant-graf for $b = 2$ med positive og negative vægtede kanter, hvor Dijkstra's algoritme beregner en forkert afstand fra s til t. Angiv for hver knude i eksemplet de korrekte afstande fra s samt de afstande Dijkstra's algoritme beregner. □

Spørgsmål c: Angiv en algoritme med udførelsestid $O(n)$, der beregner den korteste afstand fra den nedre venstre knude s til den øverste knude t i en trekant-graf med positive og negative vægtede kanter. Argumenter for udførelsestiden. □

Spørgsmål d: Angiv en algoritme, der finder to knuder u og v, $u \neq v$, med *kortest afstand* i en trekant-graf med positive og negative vægtede kanter. Angiv udførelsestiden. □

Opgave 2 (25%)

I nedenstående graf er den tungeste kant på stien (a, c, e, h) kanten (e, h) med vægt 10.



Spørgsmål a: Angiv en sti i ovenstående graf fra a til h, hvor den *tungeste kant på stien er lettest mulig.* □

Spørgsmål b: Givet to knuder u og v i en vægtet graf, angiv en algoritme der finder en sti fra u til v hvor *vægten af den tungeste kant på stien er lettest mulig.* Angiv algoritmens udførselstid som funktion af antal knuder n og antal kanter m i grafen. □

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

> *Lad G være en vægtet uorienteret graf hvor alle kanter har forskellige vægte. Lad e være en kant i G. Et minimum udspændende træ for G indeholder e hvis og kun hvis G ikke indeholder en cykel C, hvor e er den tungeste kant i cyklen C.*

Spørgsmål c: Beskriv en algoritme, der givet en vægtet graf G og en kant e i grafen, afgør om e er indeholdt i et minimum udspændende træ for G. Det antages at alle vægtene er forskellige. Udførselstiden for algoritmen skal være $O(m)$, hvor m er antal kanter i grafen. □

Opgave 3 (25%)

Givet to sekvenser $S = S[1] \cdots S[n]$ og $T = T[1] \cdots T[m]$ ønsker vi i denne opgave at finde en *længste fælles voksende delsekvens* for S og T, d.v.s. finde det største k hvor der findes $1 \leq i_1 < \cdots < i_k \leq n$ og $1 \leq j_1 < \cdots < j_k \leq m$ hvor $S[i_1] = T[j_1], \ldots, S[i_k] = T[j_k]$ og $S[i_1] < S[i_2] < \ldots < S[i_k]$.

Sekvenserne $S = \underline{2}\ 5\ 4\ \underline{3}\ \underline{4}\ \underline{6}$ og $T = 4\ \underline{2}\ \underline{3}\ 4\ 5\ \underline{4}\ \underline{6}$ har for eksempel sekvensen 2 3 4 6 som en længste fælles voksende delsekvens.

Spørgsmål a: Angiv en længste fælles voksende delsekvens for sekvenserne

$$\begin{aligned} S &= 1\ 5\ 3\ 7\ 2\ 4\ 5\ 9\ 4\ 6\ 8 \\ T &= 3\ 7\ 5\ 2\ 4\ 3\ 5\ 6\ 8\ 7 \end{aligned}$$

$\square$

For $S[i] = T[j]$ lad $L(i, j)$ betegne længden af en længste fælles voksende delsekvens for $S[1] \cdots S[i]$ og $T[1] \cdots T[j]$ som indeholder $S[i]$. For $S[i] \neq T[j]$ lader vi $L(i, j) = 0$.

Det påstås at $L(i, j)$ opfylder følgende rekursionsformel:

$$L(i,j) = \begin{cases} 0 & \text{hvis} \quad S[i] \neq T[j] \\ 1 & \text{hvis} \quad S[i] = T[j] \ \wedge \ (i = 1 \ \vee \ j = 1) \\ 1 + \displaystyle\max_{(s,t)\,:\,1 \leq s < i \,\wedge\, 1 \leq t < j \,\wedge\, S[s] < S[i]} L(s,t) & \text{hvis} \quad S[i] = T[j] \ \wedge \ i > 1 \ \wedge \ j > 1 \end{cases}$$

Spørgsmål b: Udfyld nedenstående tabel for $L(i, j)$, for $1 \leq i \leq 5$ og $1 \leq j \leq 6$, for sekvenserne

$$\begin{aligned} S &= 4\ 2\ 4\ 3\ 5 \\ T &= 2\ 3\ 2\ 5\ 4\ 5 \end{aligned}$$

$i \backslash j$	1	2	3	4	5	6
1						
2						
3						
4						
5						

$\square$

Spørgsmål c: Angiv en algoritme der beregner tabellen for $L(i, j)$, for $1 \leq i \leq n$ og $1 \leq j \leq m$, og som finder længden af en længste fælles voksende delsekvens for S og T. Angiv algoritmens udførselstid. $\square$

Spørgsmål d: Udvid algoritmen fra spørgsmål c) til at rapportere en længste fælles voksende delsekvens. Angiv algoritmens udførselstid. $\square$

(Opgavesættet fortsætter)

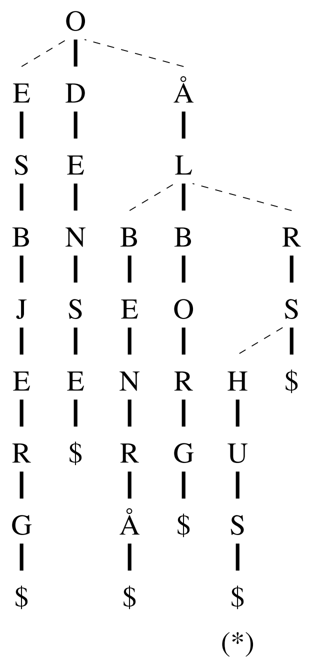
Opgave 4 (25%)

Ternære trier er en variation af trier til at gemme n strenge over et ordnet alphabet.

Hver knude gemmer et symbol og hver knude har højst tre børn: et *venstre barn*, et *midter barn*, og et *højre barn*. Når man skal søge efter en streng P og står i en knude v i en ternær trie, så sammenligner man det aktuelle tegn $P[i]$ i strengen med symbolet ved knuden v. Hvis $P[i]$ er mindre (henholdsvis større) end symbolet ved v så fortsættes søgningen efter $P[i]$ ved v's venstre barn (henholdsvis højre barn). Hvis $P[i]$ er lig med symbolet ved v, så fortsættes søgningen efter $P[i+1]$ ved v's midter barn. Således fortsættes indtil hele sekvensen er matchet.

Det antages at alle strenge afsluttes med et specielt symbol "\$", således at hvert blad svarer præcis til en streng.

I nedenstående eksempel svarer bladet $(*)$ til strengen ÅRHUS\$, da Å>O, Å=Å, R>L, R=R, H<S, H=H, U=U, S=S, og \$=\$.



Spørgsmål a: Angiv strengene der er gemt i ovenstående ternær trie. Stengene skal angives i leksikografisk rækkefølge. □

Spørgsmål b: Angiv pseudo-kode for hvordan man undersøger om en ternær trie T indeholder en streng $P = P[1] \cdots P[m]$ (hvor $P[m] = \$$). Angiv algoritmens udførselstid som funktion af m og n. □

Spørgsmål c: Angiv pseudo-kode for hvordan en streng $P = P[1] \cdots P[m]$ kan indsættes i en ternær trie T (hvor $P[m] = \$$). Angiv algoritmens udførselstid som funktion af m og n. □