Opgave 1

Givet en uordnet liste S af n sammenlignelige elementer, beskriv en effektiv algoritme til at finde de $\lceil\sqrt{n}\rceil$ elementer, som i en ordnet version af S ville ligge tættest på medianen. Hvad er udførelsestiden for din algoritme?

Opgave 2

Givet en uordnet liste S af n sammenlignelige elementer og et heltal k, beskriv en $O(n \log k)$ forventet-tids-algoritme til at finde de $O(k)$ elementer som har rang $\lceil n/k \rceil$, $2\lceil n/k \rceil$, $3\lceil n/k \rceil$, ... (dvs find hvert $\lceil n/k \rceil$te element fra S hvis vi havde sorteret S).

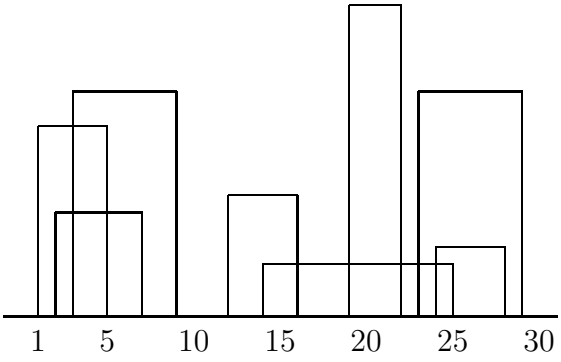
Opgave 3

Vi er givet n møtrikker og n bolte, der parvist passer sammen. For en given møtrik og bolt kan vi afgøre om bolten er for stor til møtrikken, for lille eller passer til. Det er (af en eller anden uvis grund) ikke muligt at afgøre størrelsesforholdet mellem to bolte eller to møtrikker. Lav en algoritme der parrer møtrikker og bolte. Hvad er algoritmens udførelsestid?

Opgave 4

Følgende tegning (der består af et antal rektangler) kan opfattes som repræsenterende bygninger i en by.

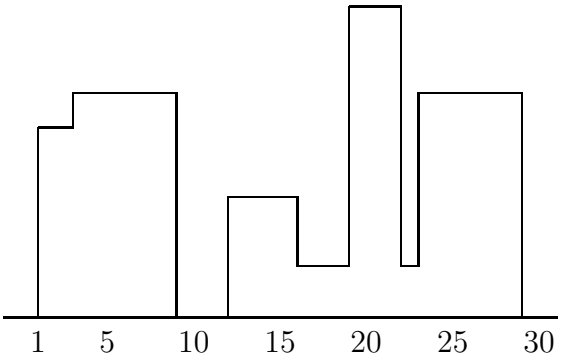
(*)



(a)

Byens *silhuet* repræsenteres af følgende tegning

(**)



(b)

Opgaven går ud på at skrive en del-og-kombiner algoritme, der læser en følge af elementer af formen (l_i, h_i, r_i), som hver angiver et rektangel, hvis venstre (højre) side har x-koordinat l_i (r_i), og hvis højde er h_i. "Byen" (*) repræsenteres således af følgende data

(1,11,5) (2,6,7) (3,13,9) (12,7,16) (14,3,25) (19,18,22) (23,13,29) (24,4,28)

Algoritmen skal producere en liste af tal af formen

(***) $(x_0, h_1, x_1, h_2, \ldots, x_{i-1}, h_i, x_i, \ldots, h_n, x_n)$

hvor x'erne er voksende og h_i angiver silhuethøjden mellem x_{i-1} og x_i. (**) repræsenteres således af følgende liste

(1,11,3,13,9,0,12,7,16,3,19,18,22,3,23,13,29)

a) Antag, at der er givet en silhuet af formen (***) samt en bygning (l, h, r). Hvordan opdateres silhuetten til også at omfatte denne bygning?

b) Udvid svaret på a) til at vise hvordan man kombinerer to silhuetter.

c) Skriv del-og-kombiner algoritmen for hele problemet og angiv dens udførelsestid.

Opgave 5

Lad $x = x_1x_2 \ldots x_n$, $y = y_1y_2 \ldots y_m$ og $z = z_1z_2 \ldots z_{n+m}$ være tre strenge af længde henholdsvis n, m og $n+m$. Vi kalder z et *flet* af x og y, hvis x og y findes som to disjunkte delsekvenser i z, og disse tilsammen udgør hele z.

> Eksempler: `gulerod` er et flet af `uro` og `gled`, og `dalgatorastritukturmerer` er et flet af `algoritmer` og `datastrukturer`.

For $0 \leq i \leq n$ og $0 \leq j \leq m$ lader vi $F[i,j]$ være en boolsk værdi, der angiver, hvorvidt strengen $z_1z_2 \ldots z_{i+j}$ er et flet af strengene $x_1x_2 \ldots x_i$ og $y_1y_2 \ldots y_j$. Her defineres $x_1x_2 \ldots x_i$ som den tomme streng, når $i = 0$ (og tilsvarende for y og z).

$F[i,j]$ kan beskrives ved følgende rekursionsformel:

$$F[i,j] = \begin{cases} X_{ij} \vee Y_{ij}\,, & i,j \geq 1 \\ X_{ij}\,, & i \geq 1, j = 0 \\ Y_{ij}\,, & i = 0, j \geq 1 \\ \texttt{Sand}\,, & i,j = 0, \end{cases}$$

hvor X_{ij} og Y_{ij} er udsagnene

$$\begin{aligned} X_{ij} &= (z_{i+j} = x_i \ \wedge \ F[i-1,j]), \\ Y_{ij} &= (z_{i+j} = y_j \ \wedge \ F[i,j-1]). \end{aligned}$$

Spørgsmål a: Opskriv tabellen for F, når x er `uro`, y er `gled` og z er `gulerod`. □

Spørgsmål b: Beskriv i form af pseudo-kode en algoritme baseret på dynamisk programmering, som i tid $O(nm)$ afgør, hvorvidt z er et flet af x og y. Der kræves et argument for, at algoritmen har den angivne kompleksitet. □

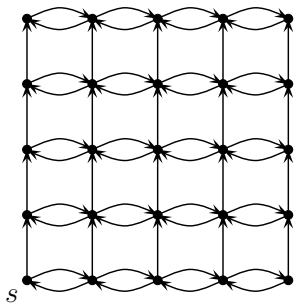
Spørgsmål c: Gør rede for, hvordan algoritmen kan udvides til i tilfælde af et positivt svar også at returnere indekserne for en delsekvens af z, som er lig x. □

Opgave 6

En *gitter-graf* er en orienteret graf hvor knuderne er arrangeret i k rækker hver indeholdende k knuder, hvor k er et positivt heltal. Lad $v_{i,j}$ betegne den $j$te knude i den $i$te række. Lad $s = v_{1,1}$. En gitter-graf har følgende knuder og kanter:

$$\begin{aligned} V &= \{v_{i,j} \mid 1 \leq i \leq k \wedge 1 \leq j \leq k\} \\ E &= \{(v_{i,j}, v_{i,j+1}) \mid 1 \leq i \leq k \wedge 1 \leq j < k\} \cup \\ &\quad \{(v_{i,j}, v_{i,j-1}) \mid 1 \leq i \leq k \wedge 1 < j \leq k\} \cup \\ &\quad \{(v_{i,j}, v_{i+1,j}) \mid 1 \leq i < k \wedge 1 \leq j \leq k\} \end{aligned}$$

Nedenstående figur viser gitter-grafen for $k = 5$.



I resten af denne opgave antager vi at alle kanter har en ikke-negativ vægt.

a) Lad n og m betegne henholdsvis antallet af knuder og kanter i en gitter-graf. Udtryk n og m som funktion af k.

b) Hvad er udførselstiden for Dijkstra's algoritme for at finde længden af de korteste veje fra $s = v_{1,1}$ til alle de øvrige knuder i en gitter-graf som funktion af k ?

c) Beskriv en algoritme der finder længden af de korteste veje fra $s = v_{1,1}$ til alle de øvrige knuder i en gitter-graf i tid $O(m)$. Argumenter for algoritmens udførselstid og korrekthed.

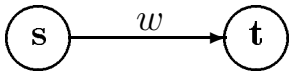
En *cylinder-graf* er en gitter-graf udvidet med ikke-negative vægtede kanter mellem den venstre og højre knude i hver række, d.v.s. E indeholder også kanterne $(v_{i,1}\,,\, v_{i,k})$ og $(v_{i,k}\,,\, v_{i,1})$ for $1 \leq i \leq k$.

d) Beskriv en algoritme der finder længden af de korteste veje fra $s = v_{1,1}$ til alle de øvrige knuder i en cylinder-graf i tid $O(m)$. Argumenter for algoritmens udførselstid og korrekthed.

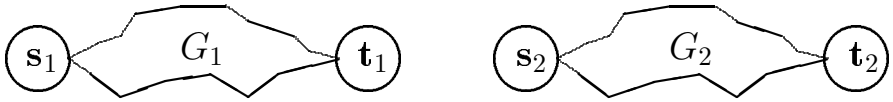
Opgave 7

Denne opgave omhandler *flade grafer*, der er specielle orienterede, vægtede grafer. De har altid to bestemte knuder **s** og **t**, og defineres i øvrigt induktivt som følger.

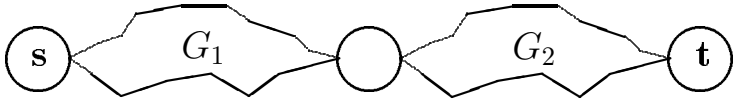
Den mindste flade graf består blot af to knuder og en enkelt kant:



To flade grafer:

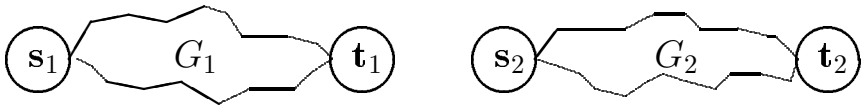


kan danne en ny ved at blive sat sammen i *serie*:

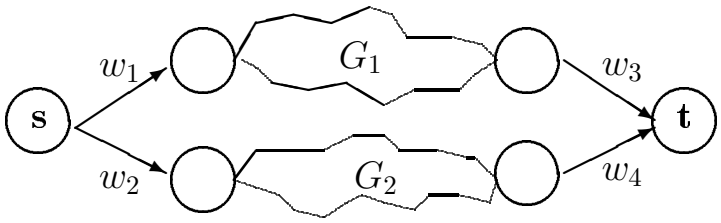


Her sammensmelter man **t**-knuden for G_1 og **s**-knuden for G_2 til en almindelig knude.

To flade grafer:



kan danne en ny ved at blive sat sammen i *parallel*:



Her introducerer man to nye knuder, der bliver henholdsvis **s**-knuden og **t**-knuden, og fire nye kanter.

a) Lad $k(n)$ være antallet af kanter i en flad graf med n knuder. Argumentér for, at $k(n) \in \mathrm{O}(n)$.

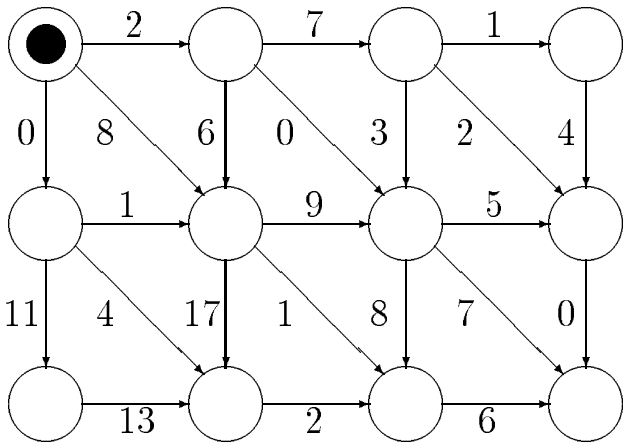
For en flad graf er vi nu interesserede i at finde længden af den korteste vej fra **s**-knuden til **t**-knuden.

b) Hvor lang tid vil det med Dijkstras algoritme tage at løse denne opgave for en flad graf med n knuder?

c) Hvordan kan man udnytte den specielle struktur af flade grafer til at opnå en mere effektiv algoritme?

Opgave 8

Et ***skråplan*** er en orienteret vægtet graf, med ikke-negative vægte, i hvilken knuderne sidder i et rektangulært gitter, og hver knude har en kant til dens eventuelle sydlige, østlige og sydøstlige nabo. Den nordvestligste knude i grafen kaldes for dens *rod*. Det følgende er et eksempel med 12 knuder, hvor roden er farvet sort.



a) Hvis et skråplan har r rækker og s søjler, så har det naturligvis $n = r \cdot s$ knuder. Hvor mange kanter har det, udtrykt ved r og s?

Vi antager i det følgende en passende repræsentation af et skråplan, der tillader, at man i konstant tid kan komme frem og tilbage mellem en knude og dens sydlige, østlige og sydøstlige naboer.

Vi er interesserede i at finde længden af den korteste vej fra roden til hver af de øvrige knuder.

b) Hvad er tidskompleksisteten for at beregne dette for et skråplan med n knuder, hvis vi benytter Dijkstras algoritme?

c) Beskriv en algoritme, der løser problemet i tid $O(n)$.

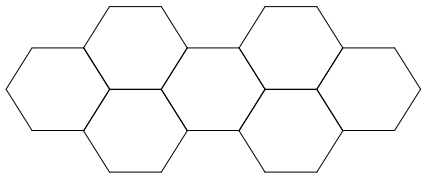
Vi er også interessede i for hvert par af knuder at finde længden af den korteste vej mellem dem.

d) Hvad er tidskompleksiteten for at beregne dette for et skråplan med n knuder, hvis vi benytter algoritmen i kapitel 7.2.1 i [GT]

e) Beskriv en algoritme, der løser problemet i tid $O(n^2)$.

Opgave 9

Betragt følgende graf — en såkaldt $H(2)$-graf —

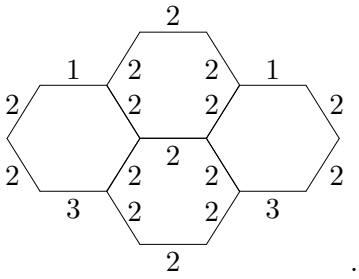


som består af 2 dobbelt-sekskanter () omgivet af 3 enkelt-sekskanter ().

Dette er et specialtilfælde af de mere generelle $H(k)$-grafer, som består af k dobbelt-sekskanter omgivet af $k+1$ enkelt-sekskanter.

Spørgsmål a: Angiv – udtrykt ved k – antallet af knuder og kanter i en $H(k)$-graf. □

Betragt nu en *vægtet* $H(k)$-graf, hvor de vandrette kanter i enkelt-sekskanterne har vægt 1 (øverst) og 3 (nederst), og hvor alle andre kanter i såvel enkelt- som dobbelt-sekskanterne har vægt 2. Eksempelvis har $H(1)$-grafen vægtet på denne måde følgende udseende



.

Spørgsmål b: Tegn en sådan vægtet $H(2)$-graf og angiv et letteste udspændende træ for grafen. □

Spørgsmål c: Angiv vægten af et letteste udspændende træ for en sådan vægtet $H(k)$-graf. Argumenter for svaret. (Vink: Husk at for en sammenhængende graf med n knuder indeholder et letteste udspændende træ $n-1$ kanter). □

Lad $N > 3$ være et vilkårligt heltal og betragt vægtede $H(k)$-grafer hvor alle øvre vandrette kanter i enkelt-sekskanterne har vægt 1 og alle de nedre vandrette kanter i enkelt-sekskanterne har vægt N (N erstatter 3), og hvor alle andre kanter i grafen har vægte der er strengt større end 1 og strengt mindre end N (vægtene kan være forskellige).

Spørgsmål d: Angiv en algoritme med udførselstid lineær i grafens størrelse, der finder et letteste udspændende træ for en sådan graf. Argumenter for at algoritmen er korrekt. Antag at grafen som sædvanligt er givet ved en *adjacency list* repræsentation. □

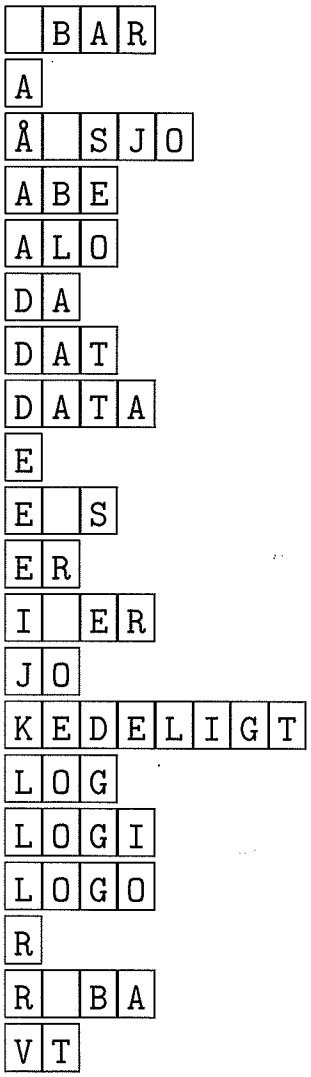
Opgave 10

På et gulv ønsker man at skrive sætninger ved hjælp af fliser med bogstaver på. Desværre kan man ikke nødvendigvis købe fliser med enkelte bogstaver på, så det kan være et problem at få skrevet de ønskede sætninger.

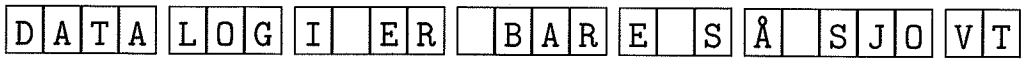
Generelt er vi givet en samling flisetyper, repræsenteret som en liste F af tekster, samt en sætning S, der blot er en tekst. Vi ønsker at afgøre, om man med (en ubegrænset forsyning af) de givne flisetyper kan skrive den valgte sætning.

For eksempel kan vi betragte sætningen:

og følgende samling af flisetyper:



Her kan problemet løses på følgende måde:



a) Benyt dynamisk programmering til at opnå en effektiv algoritme. Hvad bliver tidskompleksiteten af algoritmen?