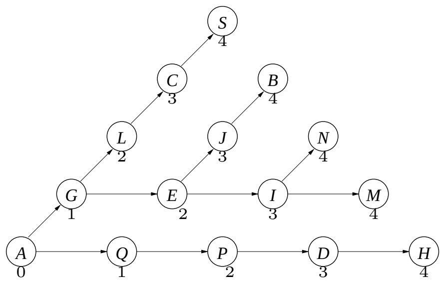
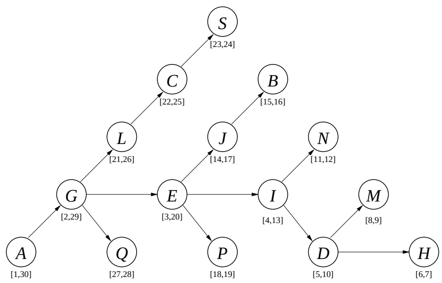
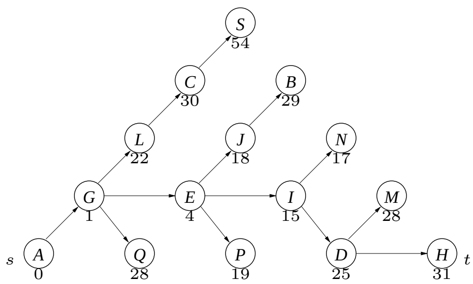
1a



1b



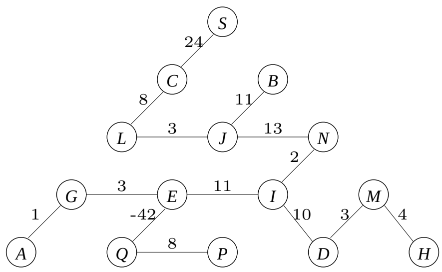
1c



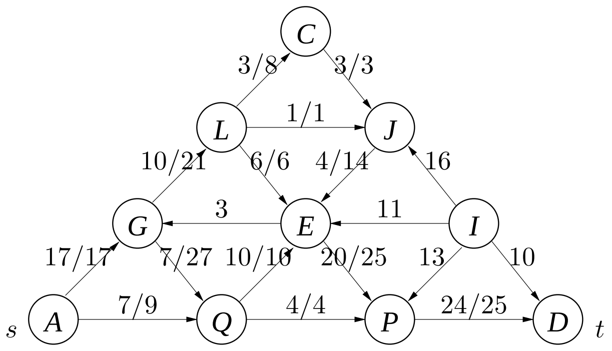
1d

En pyramide graf er en DAG med $\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = O(n^2)$ knuder og $3(\sum_{i=1}^{n} i) - 3n = O(n^2)$ kanter. Kør DAG-shortest-path på grafen i tid $O(|V| + |E|) = O(n^2)$, hvilket også bliver den totale tid.

1e



2a



Maximal strømning = 24.
Snit med kapacitet 24: $(\{s, C, G, L, Q\}, \{E, I, J, P, t\})$

2b

Sti	Forbedring
$sQPt$	4
$sQEPt$	5
$sGQEPt$	5
$sGLEPt$	6
$sGLJEPt$	1
$sGLCJEPt$	3

3a

Konstruer en graf med en knude for hver fri celle. Lav en orienteret kant til hver naboknude man kan komme til. Hvis v har straf w, så får alle kanter (u, v) vægt w. Resten af grafens kanter har vægt nul. Grafen konstrueres i tid $O(|V| + |E|)$. Kør nu Dijkstra's algoritme på grafen (med startknude s) i tid $O((|E| + |V|) \log |V|)$. Totalt bliver tiden $O((|E| + |V|) \log |V|) = O(nm \log(nm))$, da $|V| \leq nm$ og $|E| \leq 4|V| = 4nm$.

3b

Betragt grafen fra 3a) men slet kantvægtene på kanterne. Tilføj istedet kapaciter til alle *knuder* på 1, bortset fra s og t der får kapacitet $+\infty$ (bemærk at knudekapaciteter kan håndteres ved kantkapaciter ved at erstatte en knude v med to knuder v_{in} og v_{out}, lade indgående (udgående) kanter gå til v_{in} (fra v_{out}), og lave en kant $(v_{\text{in}}, v_{\text{out}})$ med knudens kapacitet). Kør nu fx. Ford-Fulkersons algoritme på grafen i tid $O(f^*|E|) = O(|E|) = O(nm)$, da der max. findes to forbedrende stier.

4a

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$C(i)$	S	F	S	F	S	F	S	S	F	S	S	S	S	F	S

4b

```
C[0]=True
for j=1 to n
  C[j]=False
  for i=1 to k
    if |S_i|<= j and C[j-|S_i|] and T[j-|S_i|+1..j]==S_i then
      C[j]=True
return C[n]
```

Tid $O(n \sum_{i=1}^{k} |S_i|)$.

4c

```
code from 4b)
report(n)

proc report(j)
  if j==0 then return
  if not C[j] then exit ‘‘No solution’’
  for i=1 to k
    if |S_i|<= j and C[j-|S_i|] and T[j-|S_i|+1..j]==S_i then
       report(j-|S_i|)
       print S_i
       return
```

Tid $O(n \sum_{i=1}^{k} |S_i|)$.

5a

$T =$ `A B C` $\underline{\texttt{B A C}}$ `B` $\underline{\texttt{C A B}}$ `B`
$S = \underline{B\,A\,C}$ forekommer på position 4 og $\bar{S}$ på position 8.

5b

Konstruer suffixtræet for $T\#\bar{T}\$$ i tid $O(n)$. Marker for hver knude v, indekserne på forekomsten længst til vestre i T og $\bar{T}$ af delstrengen s_v svarende til stien fra roden til v. Dette markeres nede fra og opefter i suffixtræet i et rekursivt gennemløb i tid $O(n)$. Hvis s_v forekommer på position i i T og j i $\bar{T}$, dvs $\bar{s_v} = T[|T| - j + 1 - |s_v| + 1..|T| - j + 1]$, så er et potentielt svar prefixet af s_v af længde d, for størst mulig d hvor $d \leq s_v$ og $i + (d - 1) < |T| - j + 1 - d + 1]$. Delstrengen der svarer til det længste potentielle svar returneres. Tid $O(n)$.