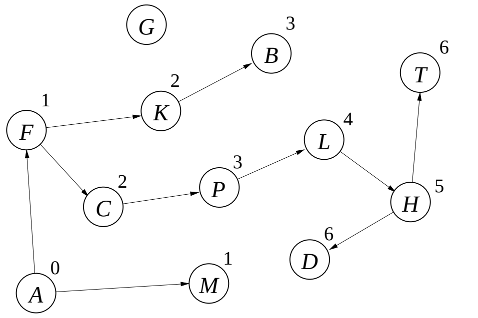
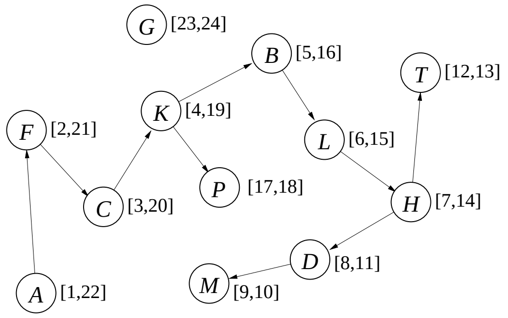
1a

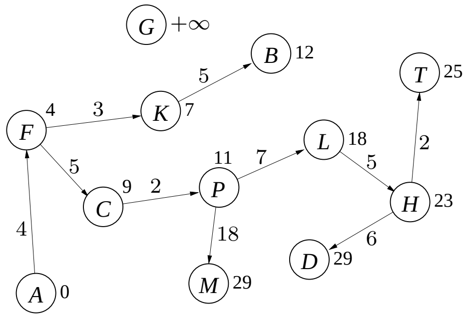


Indsættelser i Q: $A, F, M, C, K, P, B, L, H, D, T$

1b



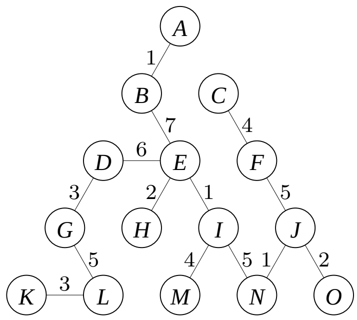
1c



1d

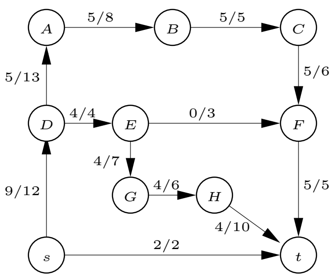
$\{A\}, \{F\}, \{G\}, \{B, C, D, H, K, L, M, P, T\}$

1e



Fjernelser fra Q: $A, B, E, I, H, M, N, J, O, F, C, D, G, L, K$

2a



Maximal strømning = 11.
Snit med kapacitet 11: $(\{s, A, B, D\}, \{C, E, F, G, H, t\})$

2b

Forbedring	Sti
2	st
3	$sDEFt$
1	$sDEGHt$
2	$sDABCFt$
3	$sDABCFEGHt$

3a

Lav et netværk med knuder $V \cup U \cup \{s, t\}$, hvor s har en orienteret kant til alle knuder i V med kapacitet 1, alle knuder u i U har en kant til t med kapacitet d_u, og for alle uorienterede kanter (v, u) i E er der en orienteret kant fra v til u med kapacitet 1. Find det maksimale flow i den resulterende graf med Ford-Fulkerson's algoritme, og returner alle kanter (v, u) hvor der sendes flow. Netværket har $n = |V| + |U| + 2$ knuder, $m = |V| + |U| + |E|$ kanter, og det maximale flow f^* er højst $|V|$. Udførselstid $O(f^* \cdot m) = O(|V|(|V| + |U| + |E|))$.

3b

Hvis en af knuderne i G har et ulige antal incidente kanter, så findes der ingen Euler tour. Ellers, lav en uorienteret todelt graf med knuder $V \cup U$, hvor hver knude i V svarer til en kant i G, og hver knude i U svarer til en knude i G. For hver orienteret kant $e = (x, y)$ i G laves en en kant (e, y), hvor $e \in V$ til $y \in U$, og for hver uorienteret kant $e = (x, y)$ i G laves kanter (e, x) og (e, y), hvor $e \in V$ og $x, y \in U$. For hver knude $u \in U$ har vi at d_u er lig antal incidente kanter til knuden u i G divideret med 2. Anvend 3a) på grafen. Der findes der en Euler tour i G, hvis og kun hvis alle knuder i V indgår i en parring. Tid $O(m^2)$, hvor m er antal kanter i G.

4a

$C(i,j)$	1	2	3	4	5
1	0	1	2	3	2
2	0	0	1	2	3
3	0	0	0	1	2
4	0	0	0	0	1
5	0	0	0	0	0

4b

```
for i=n downto 1
  for j=1 to n
    if j<=i then C[i,j]=0
    if j>i and T[i]=T[j] then C[i,j] = C[i+1,j-1]
    if j>i and T[i]!=T[j] then C[i,j] = 1+min(C[i+1,j],C[i,j-1])
return C[1,n]
```

Tid $O(n^2)$.

4c

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

proc report(i,j)
  if i=j then output T[i]
  if i<j then
     if T[i]=T[j] then output T[i]; report(i+1,j-1); output T[i]
     else
        if C[i+1,j]<=C[i,j-1] then output T[i]; report(i+1,j); output T[i]
        else output T[j]; report(i,j-1); output T[j]
```

Tid $O(n^2)$.

5a

k	forekomster af $\text{scale}_k(\texttt{abca})$
1	2, 22
2	15
3	5

5b

Løsning 1: For $k = 1, 2, 3, \ldots, \lfloor n/m \rfloor$ konstruer $P^k = \text{scale}_k(P)$, og check om P^k forekommer i T med KMP algoritmen. Total tid $O(\frac{n}{m} n) = O(n^2/m)$.

Løsning 2: For strengen T konstruer sekvensen $S = (a_1, c_1), (a_2, c_2), \ldots, (a_s, c_s)$, således at $a_i \neq a_{i+1}$, og T består af c_1 a_1-er efterfulgt af c_2 a_2-er etc. F.eks. bliver $T = \texttt{aabccca}$ til $(\texttt{a}, 2), (\texttt{b}, 1), (\texttt{c}, 3), (\texttt{a}, 1)$. For $k = 1, 2, \ldots$ søger vi intuitivt efter P i T, hvor vi erstatter k på hinanden identiske tegn i T med kun et tegn. Hvis der er færre end k identiske tegn efter hinanden, så kan disse ikke indgå i et match med $\text{scale}_k(P)$ (og heller ikke for større værdier af k). Vi fjerner derfor alle par

(a_i, c_i) fra sekvensen, hvor $c_i < k$, og der hvor vi fjerner par deler vi sekvensen S op i delsekvenser, således at hver delsekvens svarer til en delstreng hvor alle tegn forekommer mindst k gange efter hinanden. For sådan en delsekvens konstruerer vi nu en streng, hvor vi for hvert par (a_i, c_i) outputer $\lfloor c_i/k \rfloor$ forekomster af a_i til og med det første par hvor $(c_i \bmod k) \neq 0$ eller indtil vi har kigget på alle par. Hvis vi stopper ved et c_i hvor $(c_i \bmod k) \neq 0$, så starter vi en ny streng bestående af $\lfloor c_i/k \rfloor$ forekomster af a_i, og derefter betragter de efterfølgende par som ovenfor. F.eks. hvis vi har en delsekvens $(\mathtt{a}, 6), (\mathtt{c}, 3), (\mathtt{b}, 8), (\mathtt{a}, 6), (\mathtt{b}, 2), (\mathtt{c}, 8), (a, 3)$ og $k = 3$ så konstruerer vi strengene $\mathtt{aacbb}, \mathtt{bbaa}, \mathtt{cc}, \mathtt{cca}$, hvorpå vi kører KMP med P. Da antal tegn der kigges på og par i den $k$te iteration er $O(n/k)$, bliver den totale tid $O(\sum_{k=1}^{n} n/k) = O(n \log n)$. Specialtilfældet hvor alle tegn i P er lig et tegn x håndteres ved at finde det maksimale c hvor (x, c) forekommer i sekvensen. $\mathrm{scale}_k(P)$ forekommer for $k = 1, 2, \ldots, \lfloor c/m \rfloor$. Dette specialetilfælde håndteres i tid $O(n)$.