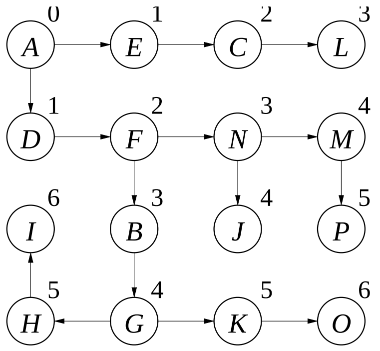
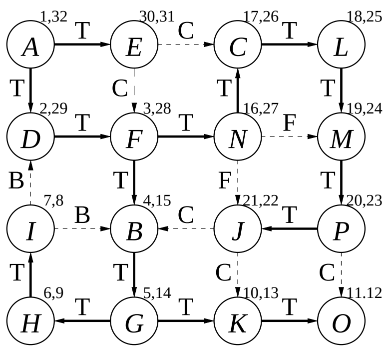
1a



Indsættelser i Q: $A, D, E, F, C, B, N, L, G, J, M, H, K, P, I, O$

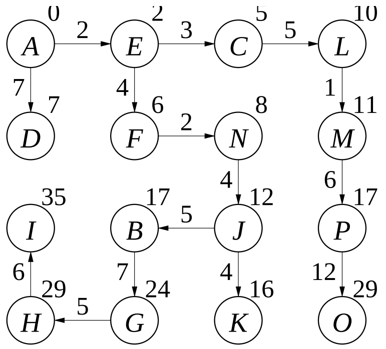
1b



1c

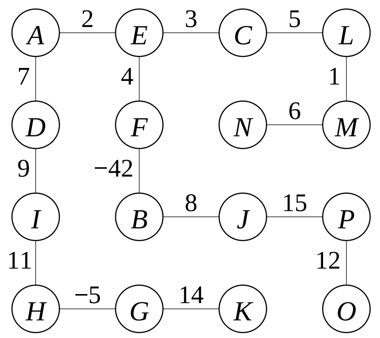
$\{A\}, \{E\}, \{K\}, \{O\}, \{C, L, D, F, N, M, I, B, J, P, H, G\}$

1d



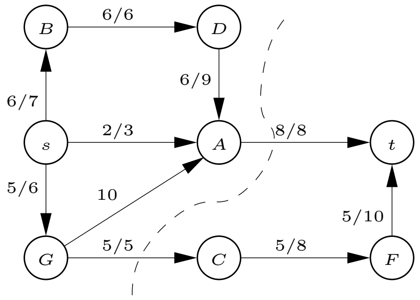
Fjernelser fra $A, E, C, F, D, N, L, M, J, K, B, P, G, O, H, I$

1e



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

2a



Maximal strømning = 13.
Snit med kapacitet 13: $(\{s, A, B, D, G\}, \{C, F, t\})$

2b

Forbedring	Sti
3	sAt
5	$sGAt$
1	$sGCFt$
4	$sBDAGCFt$

3a

Løb alle opgaverne igennem, og tæl op hvor mange gange hver maskine forekommer (som i CountingSort). Den mest hyppige maskine er flaskehalsen og frequencen er det krævede antal tidsskridt. Tid $O(J + M)$.

3b

Lad en todelt graf med knuder $U \cup V$, hvor der i U er en knude for hver opgave, og i V er en knude for hver maskine. For hver opgave j der kan udføres på maskiner m_j forbind opgaven $j \in U$ i den todelte graf til alle maskiner i $m_j \subseteq V$. Find en maksimum parring i den todelte graf. Maksimum parringen angiver hvilke maskiner vi skal udføre jobene på for at få en maksimum mængde opgaver udført i ét tidsskridt. Tid $O(|U| \cdot |E|) = O(J \sum_{j=1}^{J} |m_j|)$.

3c

Lav en graf med knuder $U \cup V \cup \{s, t\}$, hvor der i U er en knude for hver opgave j, og i V er en knude for hver maskine i. For hver maskine $i \in m_j$ forbindes knude $j \in U$ til $i \in V$ med en kant med kapacitet 1. Forbind s til alle knuder i U med kanter med kapacitet 1, og forbind alle knuder i V til t med kanter med kapacitet k. Find max flow vha. Ford-Fulkerson's algoritme. Hvis max flow f^* har værdi J, rapporteres at alle opgaver kan løses i k tidsskridt. Ellers findes der ingen løsning. Tid $O(|f^*| \cdot |E|) = O(J \sum_{j=1}^{J} |m_j|)$.

3d

Lav binær søgning i intervallet $0, \ldots, J$ efter det minimale k, hvormed 3c) finder en løsning. Dvs. 3c) skal udføres $O(\log J)$ gange. Total tid $O(\log J \cdot Jk \sum_{j=1}^{J} |m_j|)$.

4a

proc MinDist($x_1, x_2, \ldots, x_n$)
  $sum = 0$
  for $i = 1$ to n
    $sum = sum + |x_i - x_{\lceil n/2 \rceil}|$

Tid $O(n)$.

4b

$\delta(i,j)$	0	1	2	3
0	0	0	0	0
1	∞	0	0	0
2	∞	3	0	0
3	∞	4	1	0
4	∞	10	4	1
5	∞	21	10	4
6	∞	34	12	6

4c

for $i = 0$ to n
  for $j = 0$ to k
    if $i > 0$ and $j = 0$ then $\delta[i,j] = \infty$
    if $0 \le i$ and $i \le j$ then $\delta[i,j] = 0$
    else
      $\delta[i,j] = \infty$
      $M[i,j] = 0$ // used in 4d
      for $\ell = 1$ to i
        $dist = \delta[\ell-1, j-1)$+MinDist$(x_\ell, \ldots, x_i)$
        if $dist < \delta[i,j]$ then
          $\delta[i,j] = dist$
          $M[i,j] = \ell$
return $\delta(n,k)$

Tid $O(n^3k)$.

4d

code from 4c)
report(n,k)

proc report(i,j)
  if $\delta[i,j] = \infty$ then print "No solution"
  else if $i > 0$ then
    if $j \ge i$ then
      print "$x_1, \ldots, x_i$ are i singleton sets with the elements as center"
    else
      $\ell = M[i,j]$
      $c = \ell - 1 + \lceil (i - \ell + 1)/2 \rceil$
      print "x_c is center for $x_\ell, ..., x_i$"
      report$(\ell - 1, j - 1)$

Tid $O(n^3k)$.

5a

T_2 er en cyklisk rotation af T_1 hvis og kun hvis T_2 forekommer som delstreng af T_1T_1, hvilket checkes ved at køre KMP med $P = T_2$ og $T = T_1T_1$. Tid $O(n)$.

5b

Check ved hjælp af KMP om T_2 er en delstreng af strengen T, som er T_1 konkateneret med sig selv $\lceil (m + n - 1)/n \rceil$ gange. Tid $O(m)$.

5c

Konstruer suffikstræet for strengen $T_2\$T\#$. Marker i et DFS gennemløb hver knude i suffikstræet om det har et suffiks startende i T_2 og T som blad i sit undertræ. Identifier en knude v som har både et suffiks startende i T_2 og T_1' som blad i sit undertræ, og har en længst mulig streng af længde k fra roden ned til v. Hvis $k < n$, så findes der ingen cyklisk expansion af T_1 i T_2. Ellers returneres delstrengen fra roden ned til v. Tid $O(m)$.