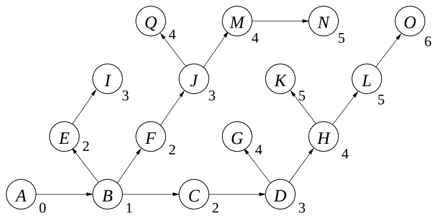
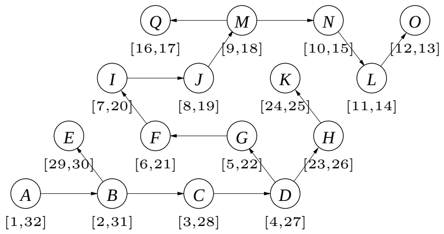
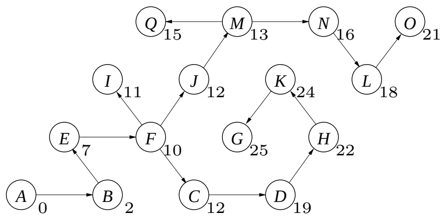
1a



1b



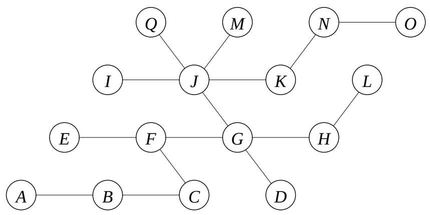
1c



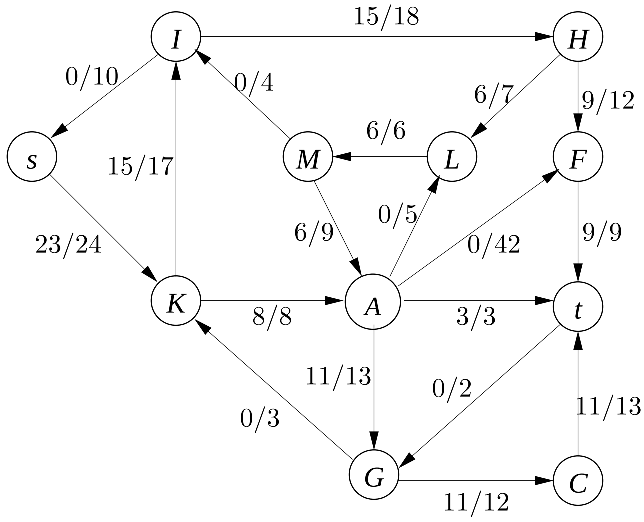
1d

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

1e



2a



Maximal strømning = 23.
Snit med kapacitet 23: $(\{s, F, H, I, K, L\},\{A, C, G, M, t\})$

2b

Sti	Forbedring
$sKAt$	3
$sKAFt$	5
$sKIHFt$	4
$sKIHFAGCt$	5
$sKIHLMAGCt$	6

3a

$T = 5$

Tid i	0	1	2	3	4	5
`Daisy`	s	e_4	e_{10}	t	t	t
`Rusty`	s	s	e_4	e_{10}	t	t
`Toby`	s	e_3	e_6	e_8	t	t
`Edward`	s	e_1	e_2	e_7	e_{11}	t

3b

Maksimal antal tog: 6

Tid i	0	1	2	3	4	5
`Daisy`	s	e_4	e_{10}	t	t	t
`Rusty`	s	s	e_4	e_{10}	t	t
`Gordon`	s	s	s	e_4	e_{10}	t
`Toby`	s	e_3	e_6	e_8	t	t
`Thomas`	s	s	e_3	e_6	e_8	t
`Edward`	s	e_1	e_2	e_7	e_{11}	t

3c

Konstruer en ny orienteret graf G' med knuder s, t, og knuder $\langle e_j, i\rangle$, hvor e_j er en kant i det oprindelige tognet G og i er en tid, $0 \leq i \leq T$. For alle par af kanter

(u, v) og (v, w) i G, dvs. kanter der har et endepunkt til fælles, lav nye kanter $(\langle(u, v), i\rangle, \langle(v, w), i+1\rangle)$ og $(\langle(v, w), i\rangle, \langle(u, v), i+1\rangle)$ for $0 \le i < T$. For alle kanter e_j og tider $0 \le i < T$ lav også kanter $(\langle e_j, i\rangle, \langle e_j, i+1\rangle)$. Endeligt for alle kanter (s, v) i G og alle tider $1 \le i \le T$, lad $(s, \langle(s, v), i\rangle)$ være en kant i G', og tilsvarende for alle kanter (v, t) i G og alle tider $0 \le i < T$, lad $(\langle(v, t), i\rangle, t)$ være en kant i G'. Lad alle *knuder*, på nær s og t, have kapacitet 1. Kør Ford-Fulkerson's max-flow algorithme (for knudekapaciteter) på G'. Hvis der er en strømning igennem en knude $\langle(u, v), i\rangle$ i G' angiver det at der et tog på kanten (u, v) til tiden i i G. En maksimal strømning i G' er det maksimale antal tog der kan sendes fra s til t i G i tiden T. Da der til hver tid højst kan komme m nye tog ud fra s, vides at den maksimale strømning er begrænset af $m \cdot T$. Da hver af de m kanter i G bliver til T kanter i G', bliver den totale tid $O((mT) \cdot (mT))$.

4a

$K(57) = 3$ og f.eks. $57 = 53 + 2 + 2$

4b

```
for i=2 to n
  if P[i]=1 then
     K[i]=1
  else
     K[i]=+infinity
     for j=2 to i-2
        if P[j]=1 and 1+K[i-j]<K[i] then
           K[i]=1+K[i-j]
return K[n]
```

Tid $O(n^2)$.

Bemærkning: Hvis Goldbach's conjecture fra 1742 viser sig at være sand (nemlig at ethvert lige tal ≥ 4 kan skrives som summen af to primtal), så kan $K(n)$ også beregnes som:

```
if P[n]=1 then return 1
if (n modulo 2=0) or (P[n-2]=1) then return 2
else return 3
```

4c

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

proc report(n)
  if P[n]=1 then
    print n;
  else
    for j=2 to n-2
      if  P[j]=1 and K[n]=1+K[n-j] then
         print j
         report(n-j)
         return
```

Tid $O(n^2)$.

5a

$T^4 = T^{10} = T^{16} =$ `aabcabaabcabaabcab`

5b

Konstruer suffixtræet for $TT\$$ i tid $O(n)$, hvor \$ er større end alle andre tegn i alfabetet. Antag at børnene til en knude er alfabetisk sorteret venstre-mod-højre m.h.t. det første tegn på kanterne ned til børnene. Følg den venstreste sti i suffixtræet indtil man når den første knude v, hvor stien fra v til roden indeholder $\geq n$ tegn. Bladene i v's undertræ svarer præcis til de suffixer af $TT\$$, hvor de n første tegn er et leksikografisk mindste skift af T. Total tid $O(n)$.