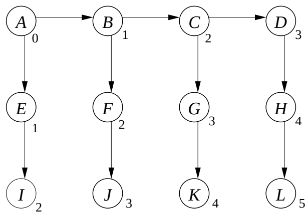
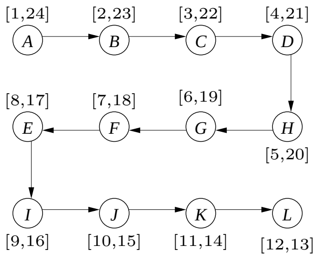
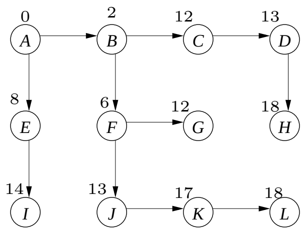
1a



1b



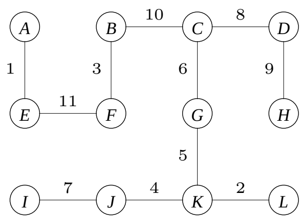
1c



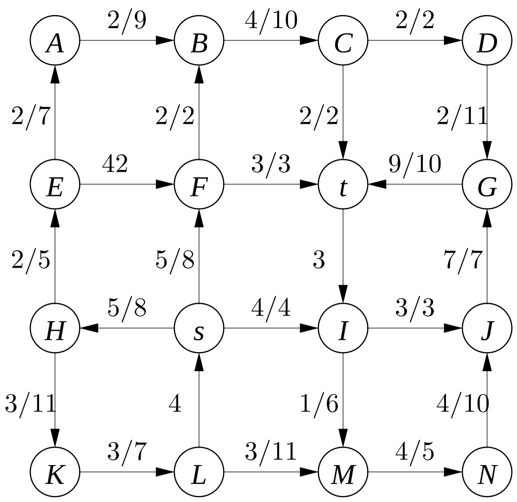
1d

En gitter-graf har $n = MN$ knuder og $m = 4MN - 2(N + M)$ kanter. Kør Dijkstra's algoritme n gange, en gang for hver knude i grafen. Husk den maksimale afstand der er fundet for nogen knuder under disse n iterationer. Total tid $O(n \cdot (m + n) \log n) = O(n^2 \log n) = O((MN)^2 \log(MN))$.

1e



2a



Maximal strømning = 14.
Snit med kapacitet 14: $(\{s, A, B, C, E, F, H, I, J, K, L, M, N\}, \{D, G, t\})$

2b

Sti	Forbedring
sFt	3
$sFBCt$	2
$sIJGt$	3
$sIMNJGt$	1
$sHEABCDGt$	2
$sHKLMNJGt$	3

3a

release(v, t)
  $t' = (t - \Delta_0(v)) \bmod (\Delta_{\text{grøn}}(v) + \Delta_{\text{rød}}(v))$
  if $t' < \Delta_{\text{grøn}}(v)$ then return t
  else return $t - t' + \Delta_{\text{grøn}}(v) + \Delta_{\text{rød}}(v)$

3b

Knude	:	S	A	B	C	T
Tidligste ankomst	:	0	10	12	6	16
Tidligste afgang	:	3	10	13	8	–

3c

Modificer Dijkstra's algoritme. For hver knude v beregn værdier v.ankomst og v.afgang. Initialiser disse til ∞, undtagen s.ankomst $= 0$ og s.afgang $=$ **release**$(s, 0)$. Lad knuder i Q i Dijkstra's algoritme have prioritet v.afgang, dvs. vi besøger knuder grådigt efter voksende afgangstider. Relax metoden ændres til:

Relax(u, v, w)
  if v.ankomst $> u$.afgang $+ w(u, v)$
    v.ankomst $= u$.afgang $+ w(u, v)$
    v.afgang $=$ **release**$(v, v$.ankomst$)$

4a

$i:$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$C_i:$	0	1	2	1	2	3	2	1	2	3	2	3	4	3	2	3	4	3	4	5	4	3	4	5	4	5

4b

```
C[0]=0
for i=1 to n
  C[i]=i
  for j=1 to k
    if x_j<=i and 1+C[i-x_j]<C[i] then
      C[i]=1+C[i-x_j]
return C[n]
```

Tid $O(nk)$.

4c

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

proc report(n)
  if n>0 then
    for j=1 to k
      if  x_j<=n and C[n]==1+C[n-x_j] then
         print x_j
         report(n-x_j)
         return
```

Tid $O(nk)$.

5a

C C A $= I_1[6..8] = I_2[1..3] = I_3[8..10] = I_4[1..3] = I_5[7..9]$ er en indikatorstreng.

5b

Konstruer suffixtræet for $I_1\#I_2\#\cdots\#I_k\#T\$$ i tid $O(n)$. Marker for hver knude v og hver i, $1 \leq i \leq k$, om der findes et blad i v's undertræ der er et suffix af I_i, og om der findes et suffix af T i v's undertræ. Dette markeres nede fra og opefter i suffixtræet i et rekursivt gennemløb, hvor markeringerne i en knude er fletningen af børnenes markeringer. Da det tager tid $O(k)$ at kigge på et barns markeringer tager dette tid $O(kn)$. For en knude v med en markering for alle I_i men ikke for T, er strengen fra roden ned til v, incl. det første tegn i v, en mulig indikatorstreng. Knuden med en kortest mulig indikatorstreng rapporteres. Total tid $O(nk)$.