

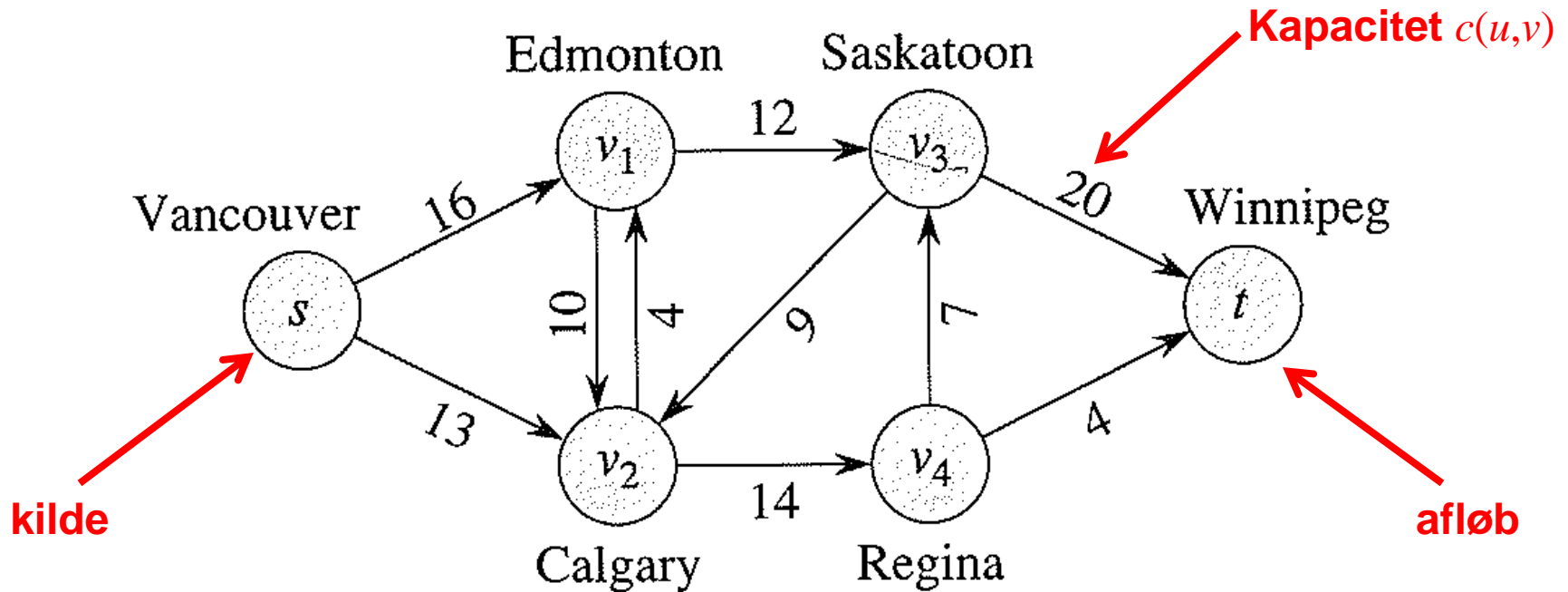
Algoritmer og Datastrukturer 2

Gerth Stølting Brodal

Maksimale Strømninger [CLRS, kapitel 26.1-26.3]



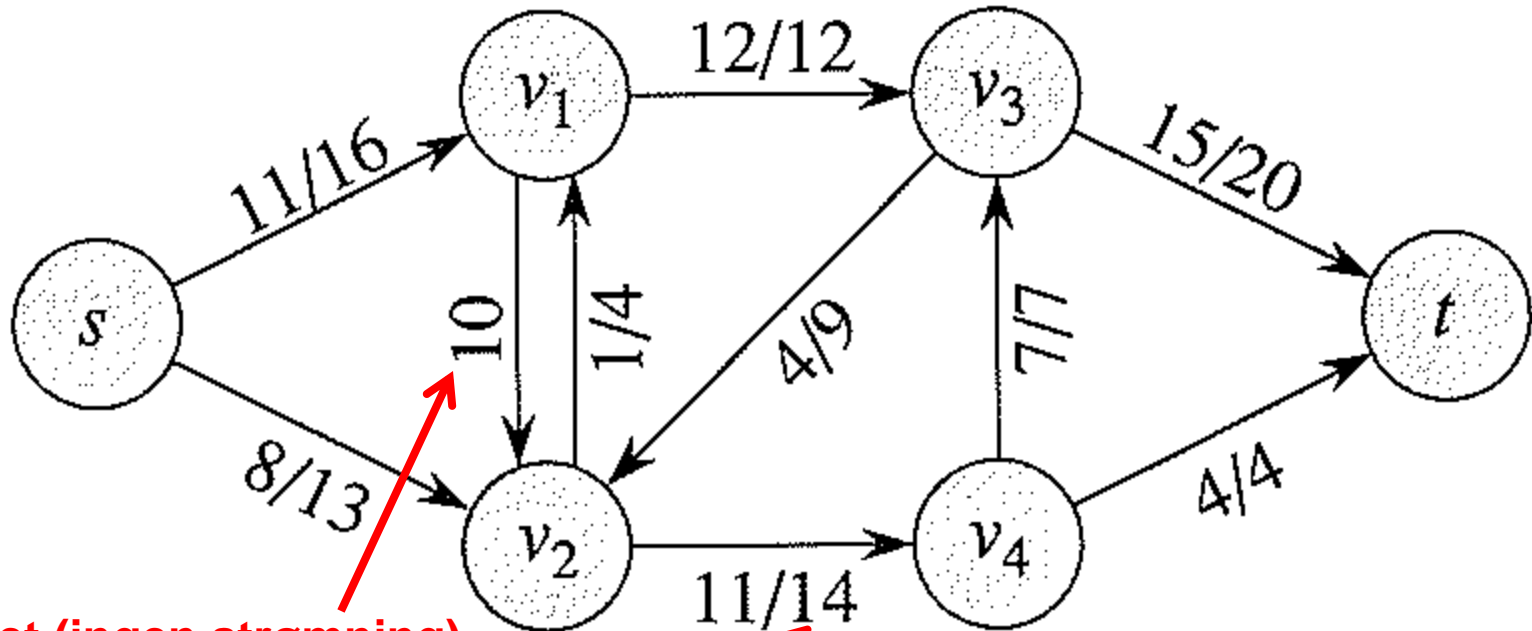
Maximale Strømninger i Netværk



Input: En orienteret graf $G=(V,E)$, hvor alle kanter (u,v) har en kapacitet $c(u,v)$, og to knuder kilde $s \in V$ (source), og afløbet $t \in V$ (sink).

Output: En maximal strømning f i netværket fra s til t .

Maximale Strømninger i Netværk



kapacitet (ingen strømning)

Strømning / kapacitet

Krav til f :

$$f(u,v) = -f(v,u)$$

$$\sum_{v \in V} f(u,v) = 0$$

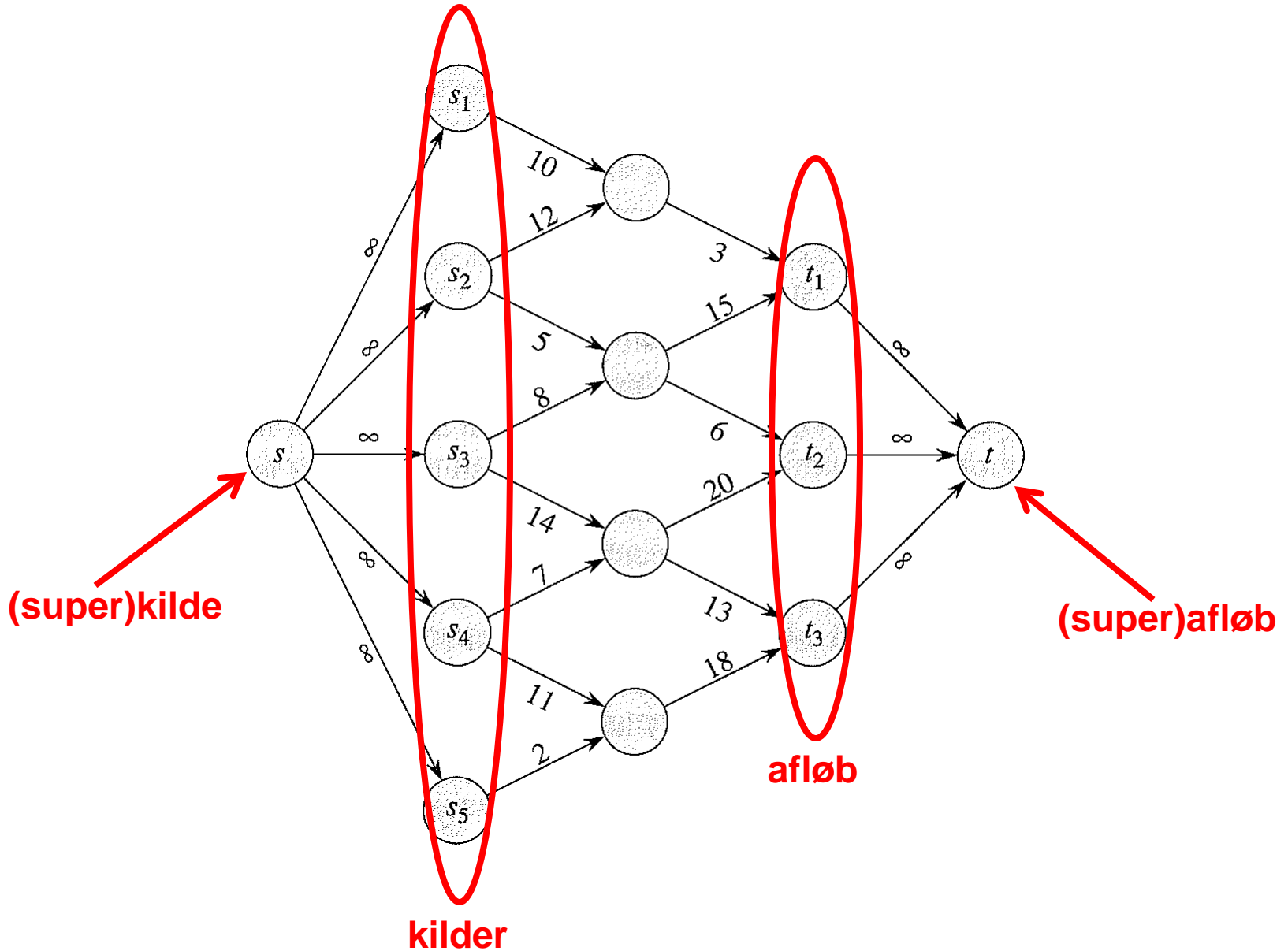
$$f(u,v) \leq c(u,v)$$

for alle kanter (strømbevaring)

for alle knuder $\neq s, t$ (strøm in=strøm ud)

for alle kanter (strømbevaring)

Flere kilder og afløb

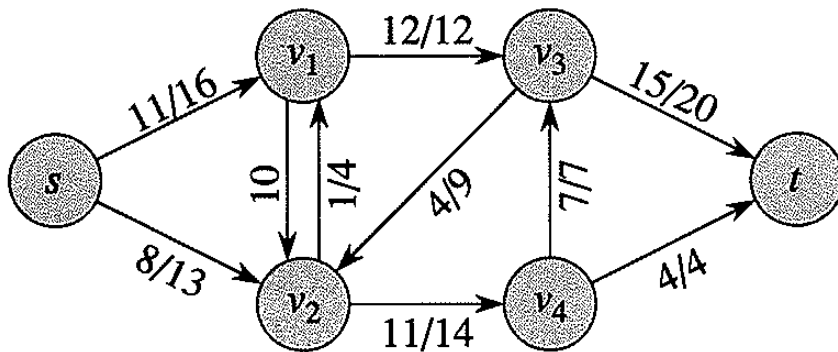


Ford-Fulkerson

FORD-FULKERSON-METHOD(G, s, t)

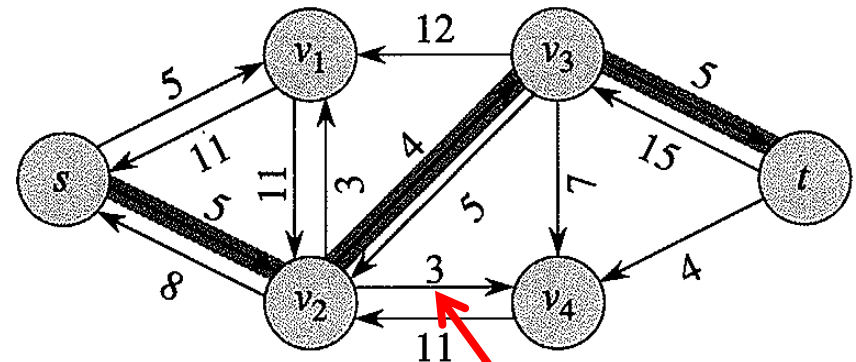
- 1 initialize flow f to 0
- 2 **while** there exists an augmenting path p in the residual network G_f
- 3 augment flow f along p
- 4 **return** f

Strømning / Kapacitet



$f(u,v) / c(u,v)$

Rest-netværk & forbedrende sti



$c_f(u,v) = c(u,v) - f(u,v)$

Observation: s - t sti i rest-netværket \equiv forbedrende sti i netværket

Ford-Fulkerson

FORD-FULKERSON(G, s, t)

1 **for** each edge $(u, v) \in G.E$

2 $(u, v).f = 0$

3 **while** there exists a path p from s to t in the residual network G_f

4 $c_f(p) = \min \{c_f(u, v) : (u, v) \text{ is in } p\}$

5 **for** each edge (u, v) in p

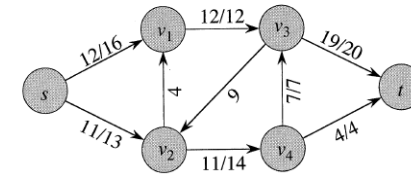
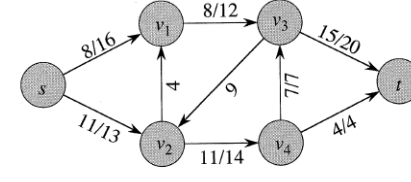
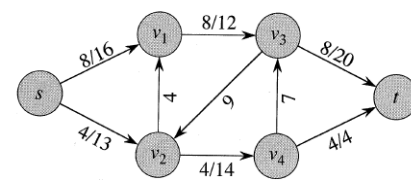
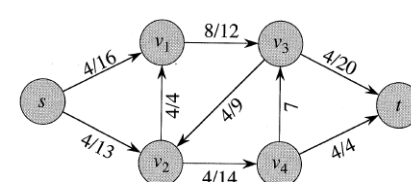
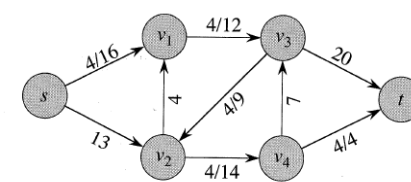
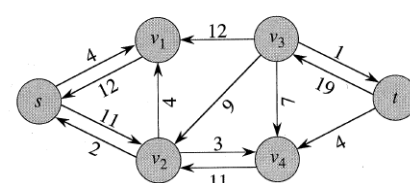
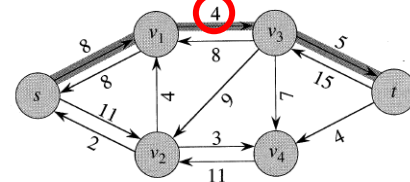
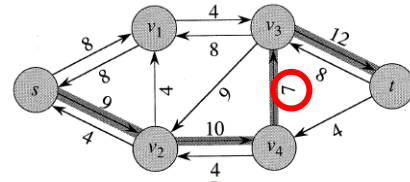
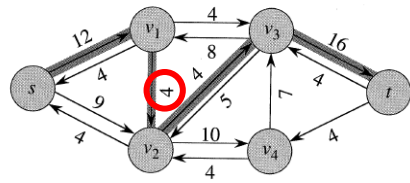
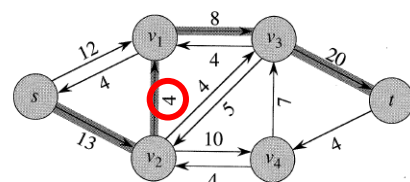
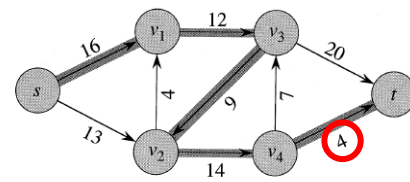
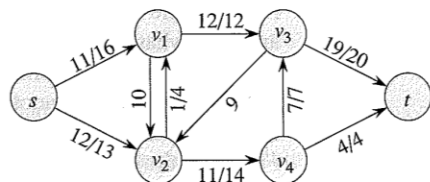
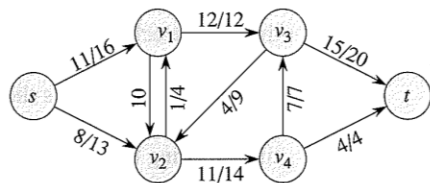
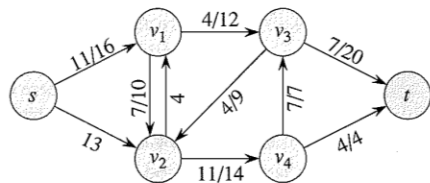
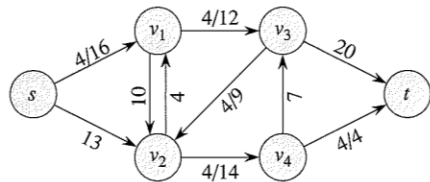
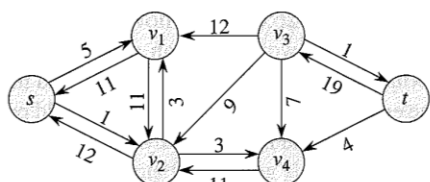
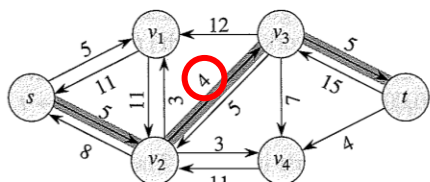
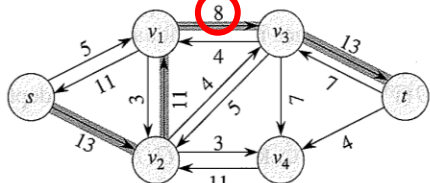
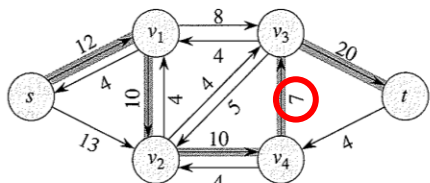
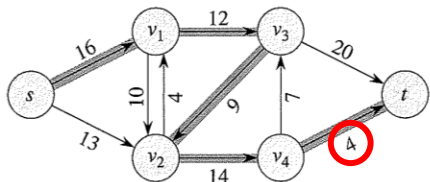
6 **if** $(u, v) \in E$

7 $(u, v).f = (u, v).f + c_f(p)$

8 **else** $(v, u).f = (v, u).f - c_f(p)$

 maximale forøgelse langs stien p

Ford-Fulkerson : eksempel



rest-netværk

aktuelle
strømning

rest-netværk

aktuelle
strømning

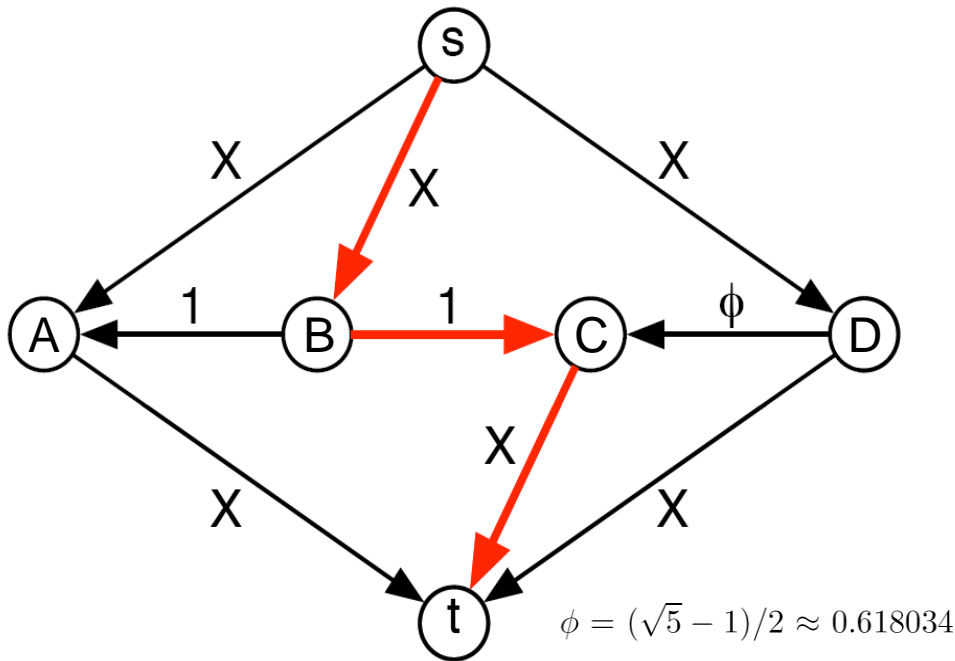
Ford-Fulkerson : Analyse

Sætning

Hvis alle kapaciteterne i et netværk er **heltallige** og f^* er en maximal strømning, så tager Ford-Fulkerson algoritmen tid $O(E \cdot |f^*|)$.

Ford-Fulkerson

∞ mange forbedrende stier



De forbedrende stier $p_0, \{p_1, p_2, p_1, p_3\}^*$ konvergerer mod en strømning af størrelse

$$1 + 2 \sum_{i=1}^{\infty} \phi^i = 1 + \frac{2}{1 - \phi} = 4 + \sqrt{5} < 7$$

hvilket er langt fra det optimale på $2X+1$.

Aktuel strømning			Forbedrende sti	Forbedring	Strømning
BA	BC	DC			
0,0000	0,0000	0,0000			0,0000
0,0000	1,0000	0,0000	sBCt	1,0000	1,0000
0,6180	0,3820	0,6180	sDCBA t	0,6180	1,6180
0,6180	1,0000	0,0000	sBCD t	0,6180	2,2361
1,0000	0,6180	0,3820	sDCBA t	0,3820	2,6180
0,6180	1,0000	0,3820	sABC t	0,3820	3,0000
0,8541	0,7639	0,6180	sDCBA t	0,2361	3,2361
0,8541	1,0000	0,3820	sBCD t	0,2361	3,4721
1,0000	0,8541	0,5279	sDCBA t	0,1459	3,6180
0,8541	1,0000	0,5279	sABC t	0,1459	3,7639
0,9443	0,9098	0,6180	sDCBA t	0,0902	3,8541
0,9443	1,0000	0,5279	sBCD t	0,0902	3,9443
1,0000	0,9443	0,5836	sDCBA t	0,0557	4,0000
0,9443	1,0000	0,5836	sABC t	0,0557	4,0557
0,9787	0,9656	0,6180	sDCBA t	0,0344	4,0902
0,9787	1,0000	0,5836	sBCD t	0,0344	4,1246
1,0000	0,9787	0,6049	sDCBA t	0,0213	4,1459
0,9787	1,0000	0,6049	sABC t	0,0213	4,1672
0,9919	0,9868	0,6180	sDCBA t	0,0132	4,1803
0,9919	1,0000	0,6049	sBCD t	0,0132	4,1935
1,0000	0,9919	0,6130	sDCBA t	0,0081	4,2016
0,9919	1,0000	0,6130	sABC t	0,0081	4,2098
0,9969	0,9950	0,6180	sDCBA t	0,0050	4,2148
0,9969	1,0000	0,6130	sBCD t	0,0050	4,2198
1,0000	0,9969	0,6161	sDCBA t	0,0031	4,2229
0,9969	1,0000	0,6161	sABC t	0,0031	4,2260
0,9988	0,9981	0,6180	sDCBA t	0,0019	4,2279
0,9988	1,0000	0,6161	sBCD t	0,0019	4,2299
1,0000	0,9988	0,6173	sDCBA t	0,0012	4,2310
0,9988	1,0000	0,6173	sABC t	0,0012	4,2322
0,9995	0,9993	0,6180	sDCBA t	0,0007	4,2330
0,9995	1,0000	0,6173	sBCD t	0,0007	4,2337
1,0000	0,9995	0,6178	sDCBA t	0,0005	4,2341
0,9995	1,0000	0,6178	sABC t	0,0005	4,2346
0,9998	0,9997	0,6180	sDCBA t	0,0003	4,2349
0,9998	1,0000	0,6178	sBCD t	0,0003	4,2352
1,0000	0,9998	0,6179	sDCBA t	0,0002	4,2353
0,9998	1,0000	0,6179	sABC t	0,0002	4,2355
...

Edmonds-Karp Algoritmen

= Ford-Fulkerson hvor forbedrende stier har færrest mulige kanter (BFS)

Sætning Finder $O(V \cdot E)$ forbedrende stier, dvs. tid $O(V \cdot E^2)$

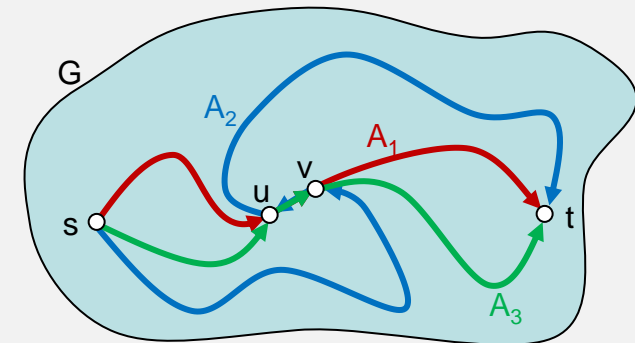
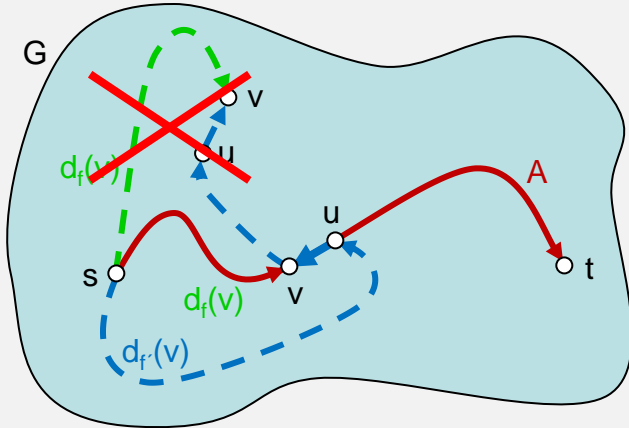
Bevis

f = flow i G

A = forbedrende sti i G_f

f' = flow efter A

$d_f(v)$ = afstand (#kanter) fra s til v i G_f



Lemma 1 $\forall v : d_{f'}(v) \leq d_f(v)$

Bevis Antag modsætnings der findes en knude v med $d_{f'}(v) > d_f(v)$, og v har mindst mulig $d_{f'}(v)$.

$$d_{f'}(v) = d_{f'}(u) - 1 \leq d_f(u) - 1 \leq d_f(v) - 2 \quad \text{⚡} \quad \square$$

BFS (Edmonds-Karp)
minimalitet af v
def. af $d_{f'}$

Lemma 2 En kant (u,v) kan kun være flaskehalsen på en forbedrende sti $\leq |V|/2$ gange

Bevis Antag (u,v) flaskehalsen når A_1 og senere A_3 anvendes; inden A_3 må der findes A_2 der sender flow langs (v,u) . f_i flow før A_i .

$$d_{f_3}(v) = d_{f_3}(u) + 1 \geq d_{f_2}(u) + 1 = d_{f_2}(v) + 2 \geq d_{f_1}(v) + 2$$

BFS
Lemma 1
BFS
Lemma 1

Lemma følger fra $0 \leq d_{f_i}(v) \leq |V| - 2$ for $v \notin \{s, t\}$ \square

Korollar #forbedrende stier er $\leq 2|E| \cdot |V|/2$

Maksimale strømninger – Historisk overblik

year	discoverer(s)	bound
1951	Dantzig [11]	$O(n^2mU)$
1956	Ford & Fulkerson [17]	$O(nmU)$
1970	Dinitz [13] Edmonds & Karp [15]	$O(nm^2)$
1970	Dinitz [13]	$O(n^2m)$
1972	Edmonds & Karp [15] Dinitz [14]	$O(m^2 \log U)$
1973	Dinitz [14] Gabow [19]	$O(nm \log U)$
1974	Karzanov [36]	$O(n^3)$
1977	Cherkassky [9]	$O(n^2m^{1/2})$
1980	Galil & Naamad [20]	$O(nm \log^2 n)$
1983	Sleator & Tarjan [46]	$O(nm \log n)$
1986	Goldberg & Tarjan [26]	$O(nm \log(n^2/m))$
1987	Ahuja & Orlin [2]	$O(nm + n^2 \log U)$
1987	Ahuja et al. [3]	$O(nm \log(n\sqrt{\log U}/m))$
1989	Cheriyān & Hagerup [7]	$E(nm + n^2 \log^2 n)$
1990	Cheriyān et al. [8]	$O(n^3 / \log n)$
1990	Alon [4]	$O(nm + n^{8/3} \log n)$
1992	King et al. [37]	$O(nm + n^{2+\epsilon})$
1993	Phillips & Westbrook [44]	$O(nm(\log_{m/n} n + \log^{2+\epsilon} n))$
1994	King et al. [38]	$O(nm \log_{m/(n \log n)} n)$
1997	Goldberg & Rao [24]	$O(\min(n^{2/3}, m^{1/2})m \log(n^2/m) \log U)$

[CLRS 26.2]

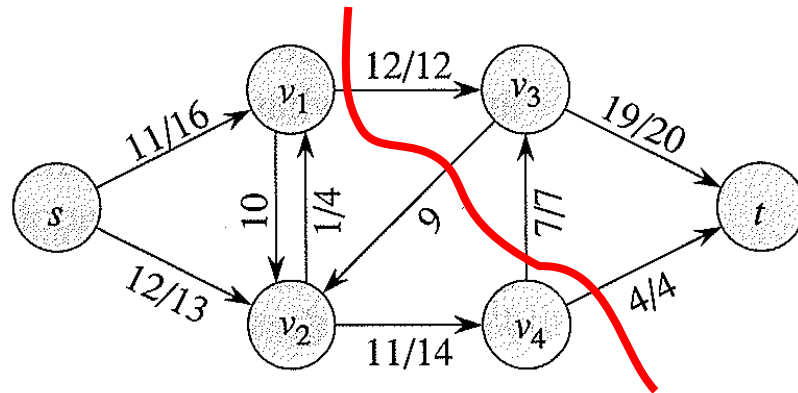
Andrew Goldberg, 1998

$n = \#$ knuder, $m = \#$ kanter, kapaciteter i intervallet $[1..U]$

Maximale strømninger

Sætning

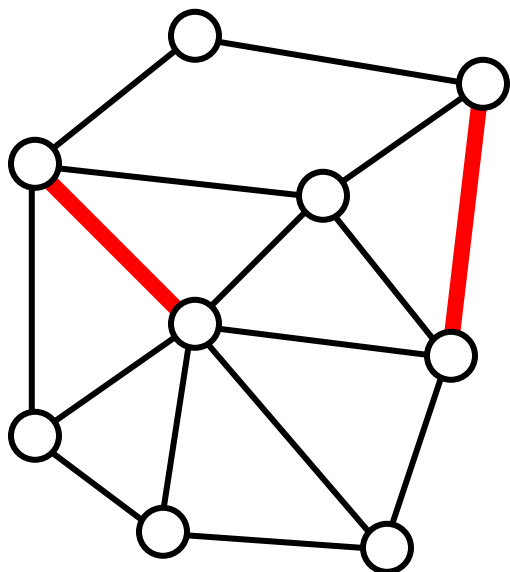
Maximal strømning = kapaciteten af et minimalt snit



Sætning

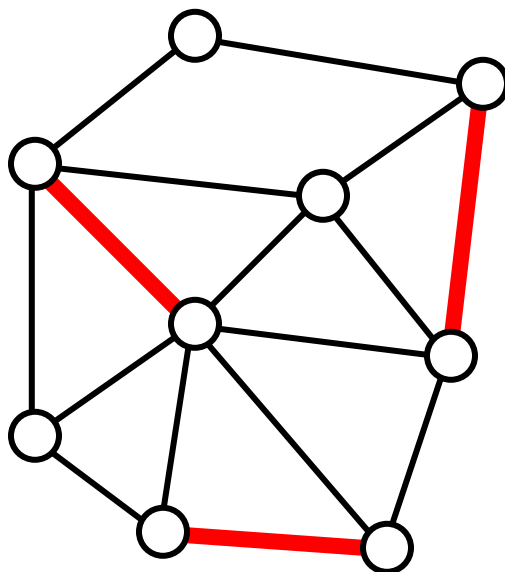
Hvis alle kapaciteter er heltallige så finder Ford-Fulkerson og Edmonds-Karp algoritmerne en strømning hvor *strømmen langs alle kanter er heltalligt*

Parringer



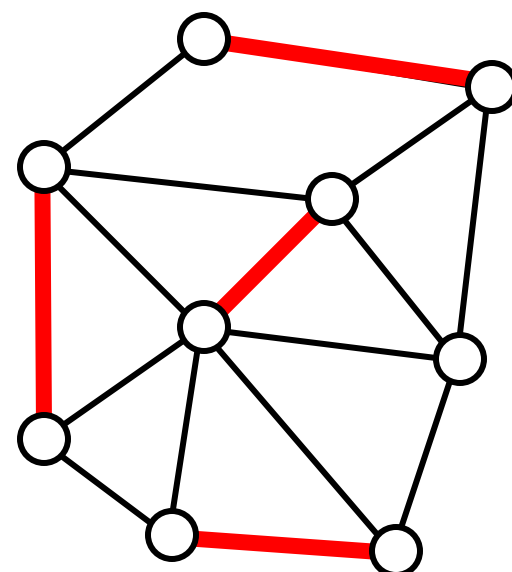
Parring

(en delmængde af kanterne hvor hver knude indgår max én gang)



Maximal Parring

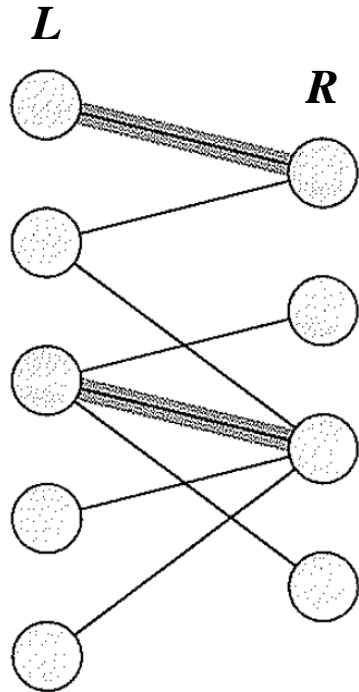
(parring hvor ingen kanter kan tilføjes – kan findes v.h.a grådige algoritme)



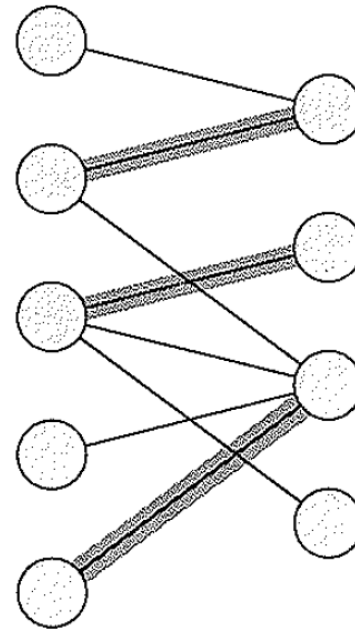
Maximum Parring

(findes ingen parringer med flere kanter)

Parringer i todelte grafer

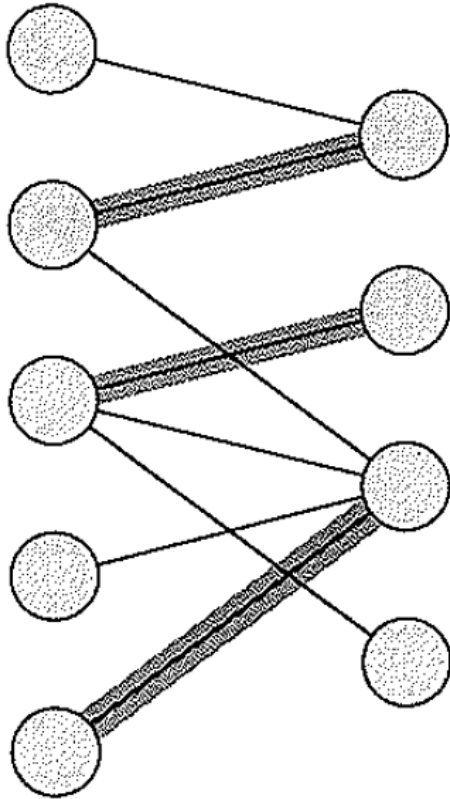


**Maximal Parringer
af størrelse 2**

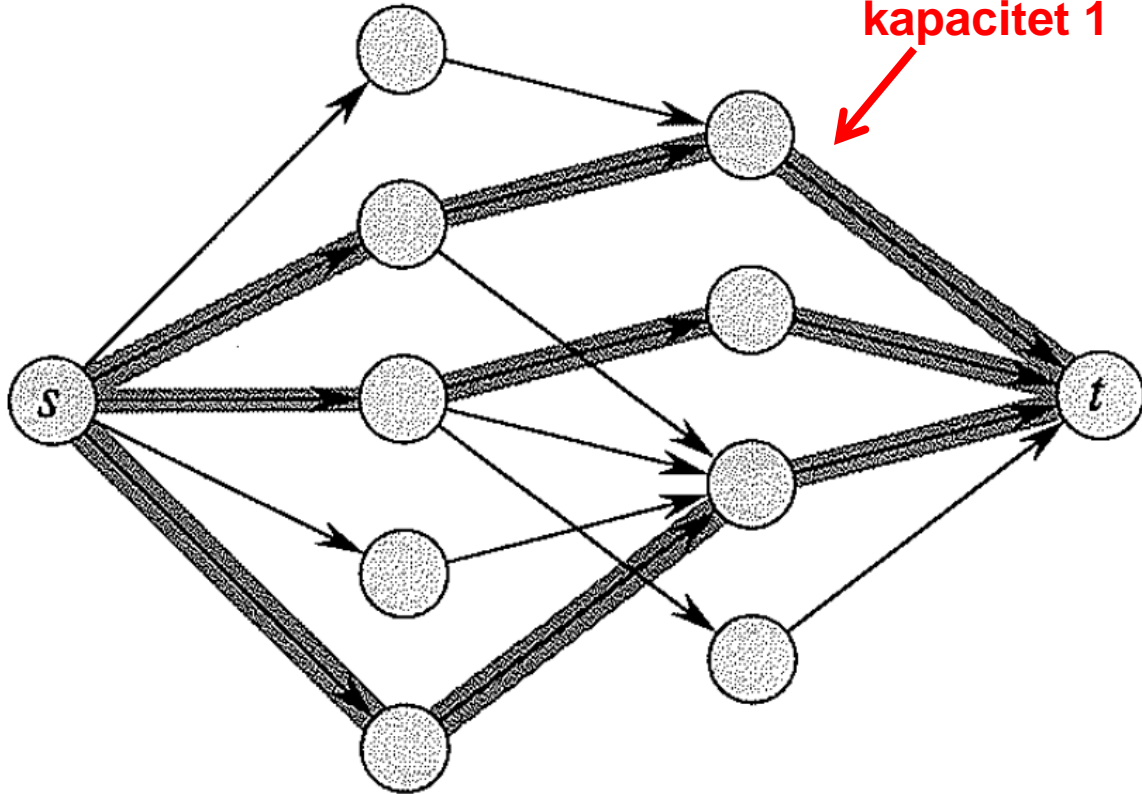


**Maximum Parringer
af størrelse 3**

Parring vs. Strømning



Todelt graf
Maximum matching



Strømnings netværk
Maximum strømning
(i en heltallig løsning,
f.x. Ford-Fulkerson)



NWERC 2007

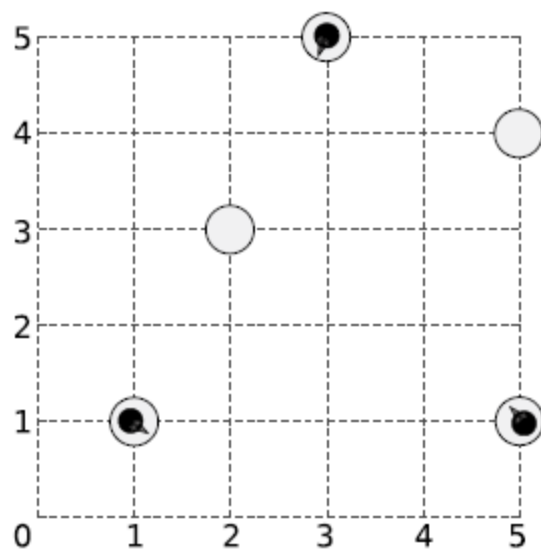
*The 2007 ACM Northwestern European Programming Contest
Utrecht University, The Netherlands*



Problem B: March of the Penguins

B March of the Penguins

Somewhere near the south pole, a number of penguins are standing on a number of ice floes. Being social animals, the penguins would like to get together, all on the same floe. The penguins do not want to get wet, so they have use their limited jump distance to get together by jumping from piece to piece. However, temperatures have been high lately, and the floes are showing cracks, and they get damaged further by the force needed to jump to another floe. Fortunately the penguins are real experts on cracking ice floes, and know exactly how many times a penguin can jump off each floe before it disintegrates and disappears. Landing on an ice floe does not damage it. You have to help the penguins find all floes where they can meet.



A sample layout of ice floes with 3 penguins on them.

Input

On the first line one positive number: the number of testcases, at most 100. After that per testcase:

- One line with the integer N ($1 \leq N \leq 100$) and a floating-point number D ($0 \leq D \leq 100\,000$), denoting the number of ice pieces and the maximum distance a penguin can jump.
- N lines, each line containing x_i, y_i, n_i and m_i , denoting for each ice piece its X and Y coordinate, the number of penguins on it and the maximum number of times a penguin can jump off this piece before it disappears ($-10\,000 \leq x_i, y_i \leq 10\,000$, $0 \leq n_i \leq 10$, $1 \leq m_i \leq 200$).

Output

Per testcase:

- One line containing a space-separated list of 0-based indices of the pieces on which all penguins can meet. If no such piece exists, output a line with the single number -1 .

Sample in- and output

Input	Output
2	1 2 4
5 3.5	-1
1 1 1 1	
2 3 0 1	
3 5 1 1	
5 1 1 1	
5 4 0 1	
3 1.1	
-1 0 5 10	
0 0 3 9	
2 0 1 1	