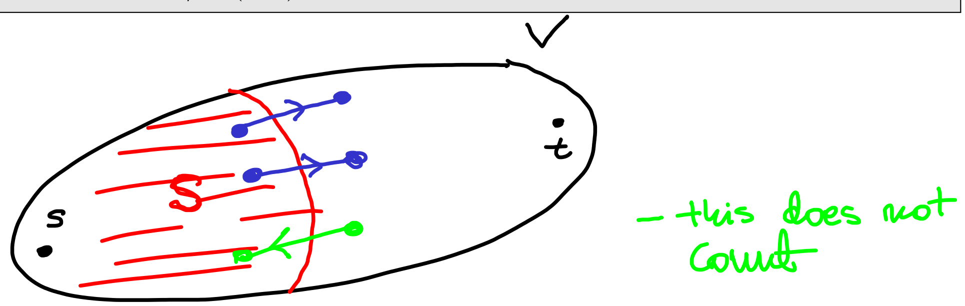
Why the flow is maximal (if there are no AP)?

Definition. A cut is any $S \subseteq V$ such that $s \in S$, $t \notin S$. The capacity of the cut S is defined as

$$c(S) = \sum_{x \in S. y \notin S, (x,y) \in A} u_{(x.y)}.$$

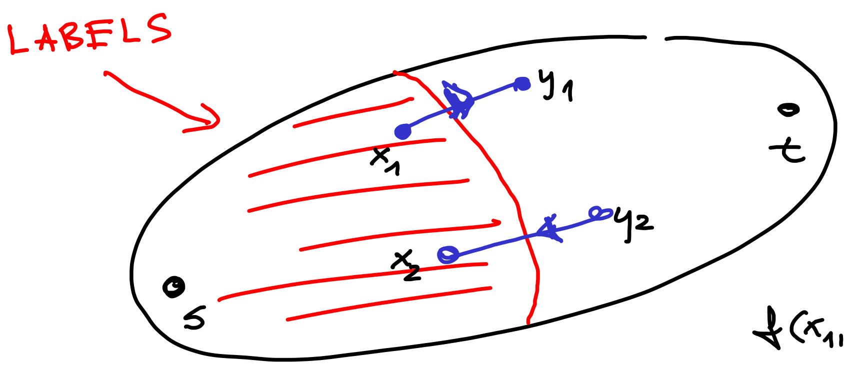


Theorem. We have $vol(f) \leqslant c(S)$ for every feasible flow f and every cut S.
If f and S satisfy $vol(f) = c(S)$ then the flow f is maximal.

If there are no AP. . .

> **Theorem.** If the labelling algorithm finds no augmenting paths then the given flow is maximal.

Proof. If LA stops finding no augmenting paths then we examine the set $E \subseteq V$ that got the labels. Then $s \in E$, $t \notin E$ so E is a cut. We check that $vol(f) = c(E)$:



$$f(x_1, y_1) = c_{(x_1, y_1)}$$

$$f(y_2, x_2) = 0$$

Does the whole algorithm works?

Lemma. If the initial data are integer-valued and we start from the initial flow with integer values then all the values of augmented flows remain integer.

Theorem. If all the capacities are integers then the Ford-Fulkerson algorithm stops after a finite number of steps.

Proof. Let

$$M = \sum_{x \in \mathrm{Out}(s)} c_{(s,x)}.$$

Then M is an upper bound of the volume of any feasible flow. At each step we augment the given flow by at least 1 so there will be no more then M steps.

Remark. We then perform at most $2M|A|$ operations during the whole process.

Corollary. If $u : A \to \mathbb{Q}_+ \cup \{\infty\}$ then FF stops after a finite number of steps.

Funny, isn't it?

The last fact is not true for in case of real-valued capacities.

$x_{n+1} + x_{n+2} = x_n \qquad x_0 = 1$

$x_n = q^n. \qquad q^{n+2} + q^{n+1} = q^n \quad / q^n$

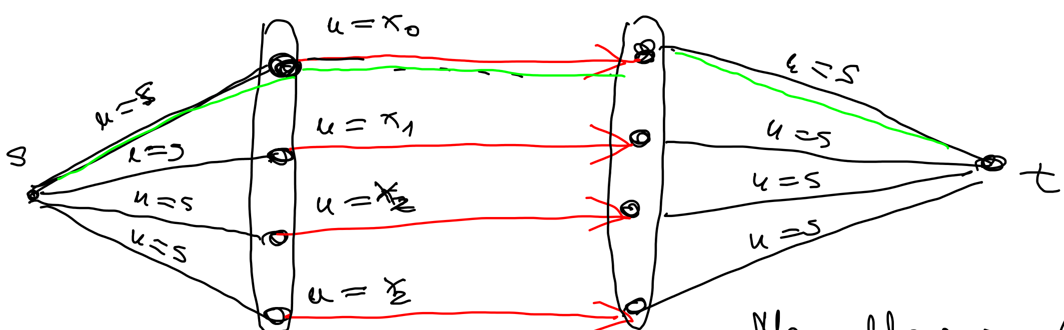
$q \neq 0$

$q^2 + q = 1$

$q^2 + q - 1 = 0 \qquad q = \frac{\sqrt{5}-1}{2} \notin \mathbb{Q}$

$0 < q < 1$

$s = \sum_{n=0}^{\infty} q^n.$



Max flow $= 4s$

arc	u	0 step	↓	↓ $a_{2,1} - a_1 - a_3$
a_1	x_0	$\cancel{x_0}$	$\cancel{x_0}$	$x_0 - x_3$
a_2	x_1	0	x_2	$x_2 + x_3 = x_1$
a_3	x_2	0	x_2	$x_2 - x_3$
a_4	x_2	0	0	0

res. value = capacity of the arc − the flow.

residual values

	0	1
a_1	0	x_3
a_2	x_1	0
a_3	x_2	x_3
a_4	x_2	x_2

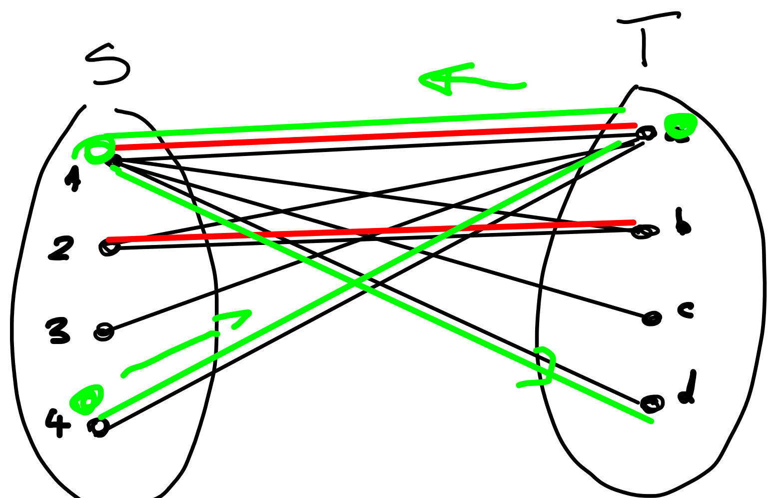
Claim. Having res. values $0, x_n, x_{n+1}, x_{n+1}$ we can get $0, x_{n+1}, x_{n+2}, x_{n+2}$

Matching in bipartite graphs

Suppose that we have a graph $G = (V, A)$ where $V = S \cup T$ and every arc in A is of the form (x, y), where $x \in S$ and $y \in T$.

A matching in such a graph is an injective function $g : D \to T$ where $D \subseteq S$ and $(x, f(x)) \in A$ for every $x \in D$.

> **Problem.** Given a bipartite graphs, find a maximal matching in it; the one maximizing $|D|$.



New matching:
4 – a
2 – b
1 – d

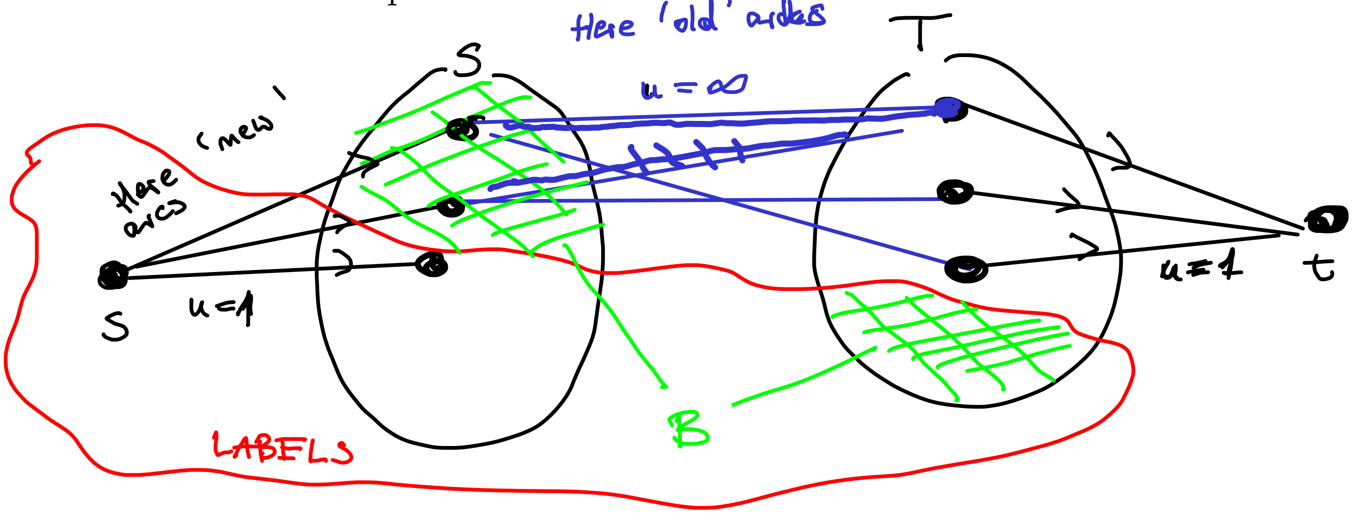
~~1 – a~~

If you match 1 with a, 2 with b, then you get stuck!

Alternating chain

Matchings from flows

Given a bipartite graph $G = (V, A)$, $V = S \dot\cup T$ etc. extend it to $\widetilde{G}$ as in the picture:



> **Observation.** Every feasible flows in $\widetilde{G}$ defines a matching in G. Augmenting paths and labelling can be interpreted inside G.

Dictionary

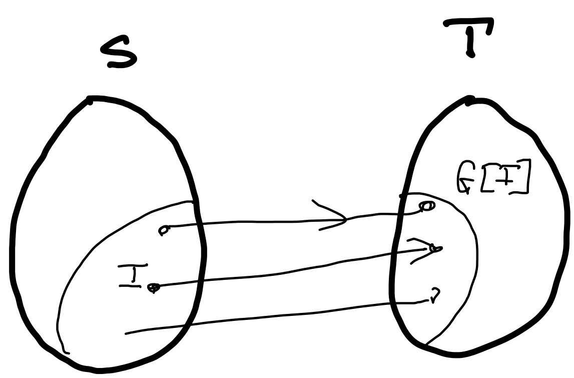
augmenting path $\longrightarrow$ alternating chain

cut L $\longrightarrow$ blocking set

$B = (S \setminus L) \cup (T \cap L)$

Algorithms prove theorems

Theorem (König's). In a bipartite graph, the size of a maximal matching equals the minimal number of blocking vertices ($B \subseteq V$ is blocking if every arc either starts in B or ends in it).



Hall's marriage theorem. In a bipartite graph $G = (V, A)$, where $G = S \cup T$ there is a matching of maximal size $|S|$ if and only if for every $I \subseteq S$ we have $|G[I]| \geqslant |I|$ (here $G[I] = \{y \in T : (x, y) \in A \text{ for some } x \in I\}$).