BJG 3.22 $N = (V \cup \{s,t\}, A, \ell \equiv 0, u)$

h height function wrt N and X

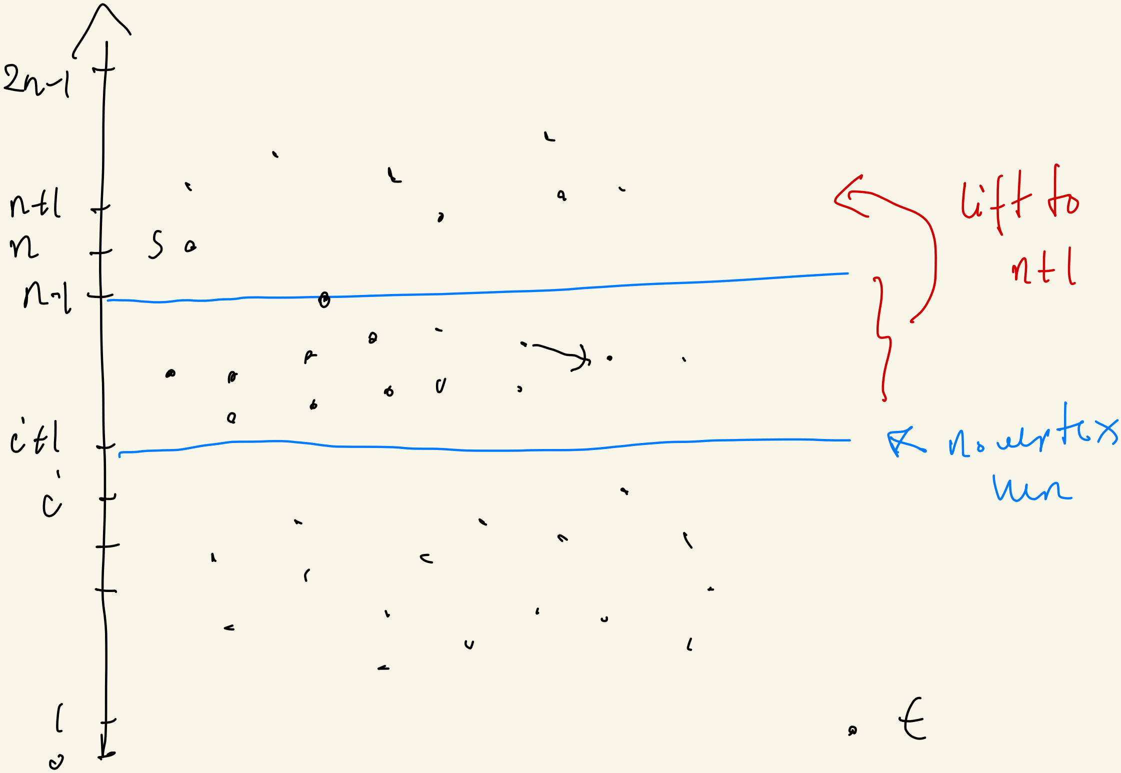
h has a hole at position $i+1$ for some $i < n$ if

$$|\{v \mid h(v) = j\}| > 0 \quad \forall j \leq i$$

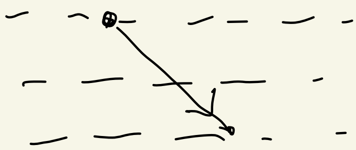
$$|\{v \mid h(v) = i+1\}| = 0$$

Define h' as

- $h'(v) = h(v)$ if $h(v) \in \{0, 1, \dots, i\} \cup \{n, n+1, \dots, 2n-1\}$
- $h'(v) = n+1$ if $i < h(v) < n$



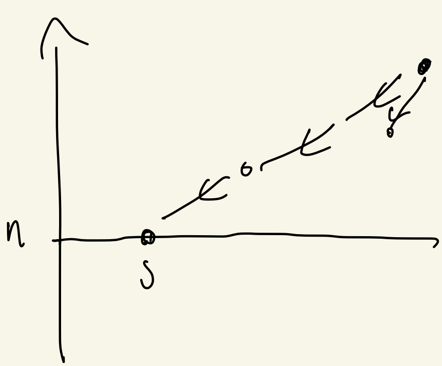
(a) show that h' is a height function



(b) show how to implement this change

h	0	1	2		i	i+1		n	n+1	2n-1
#	1	4	5		3			1		

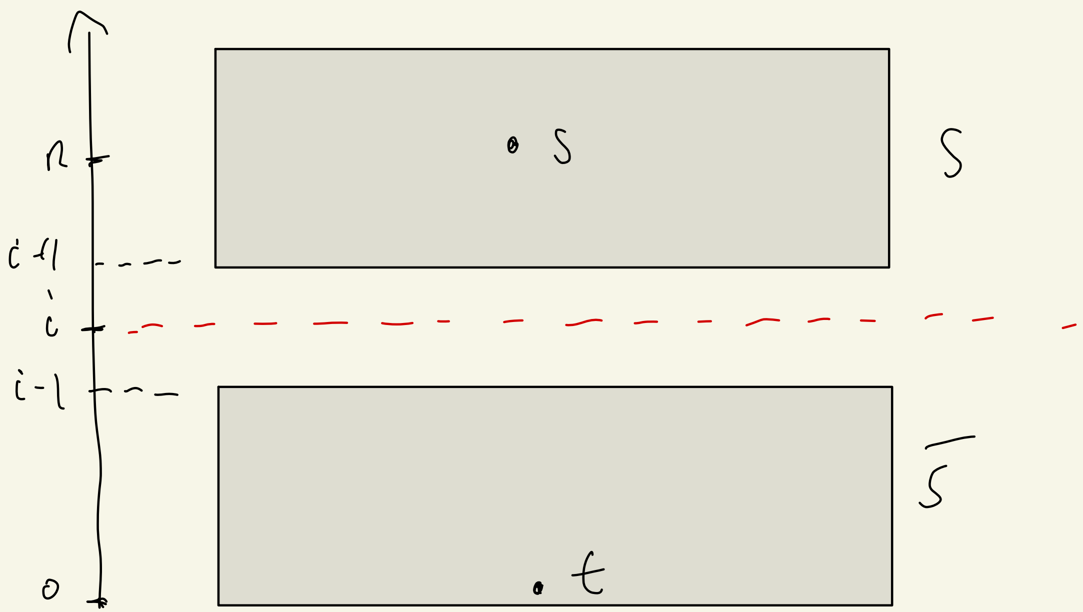
(c) Explain why changing h to h' may speed up algorithm

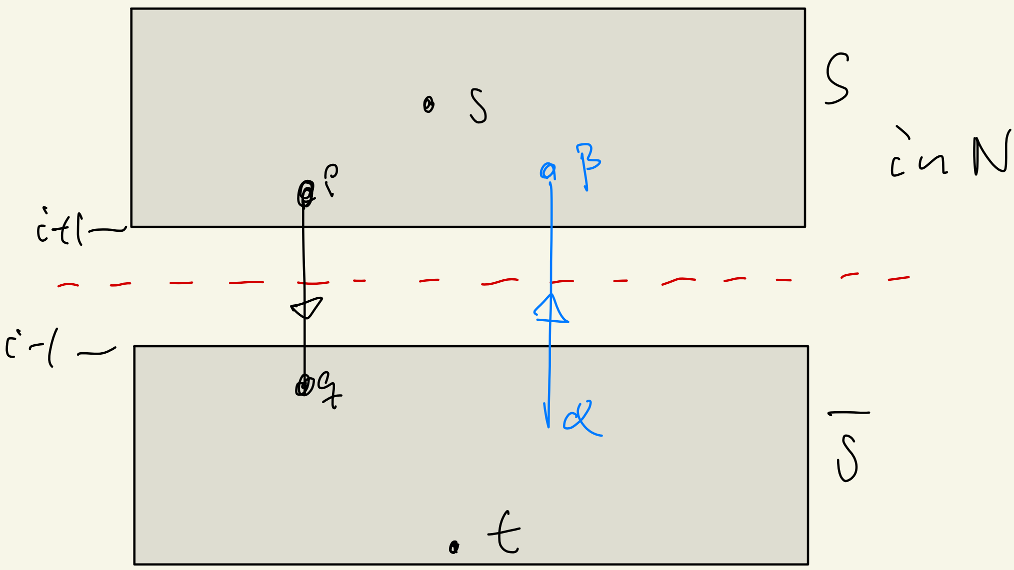


BJG 3.23

x max flow found by the preflow-push algorithm. Show how to use the height function to detect a minimum (s,t)-cut in time $O(n)$

$h(t)=0$, $h(s)=n$ so $n-2$ vertices to cover the heights in $[1,n-1]$ $\Rightarrow$ $\exists\, i \in [1,n-1]$: no vertex of height i





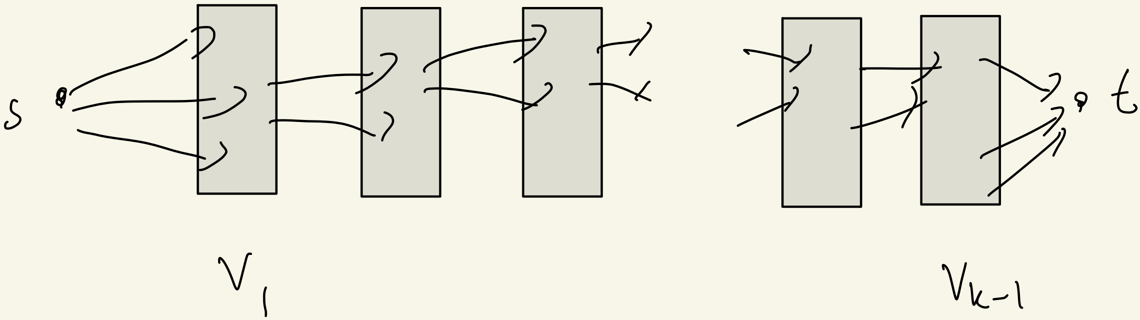
$x_{pq} = u_{pq}$ since h is a height function during the algorithm

$x_{\alpha\beta} = 0$ same reason as above

$$|x| = x(S,\bar{S}) - x(\bar{S},S)$$
$$= u(S,\bar{S})$$

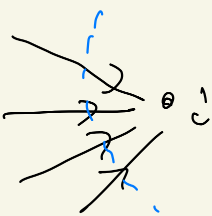
BJG 3.25   MKM-algorithm

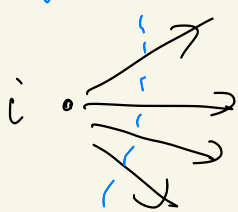
$\mathcal{D}$:



y feasible (s,t)-flow in $\mathcal{D}$ which is not blocking

$\forall i \neq s,t$: $\alpha_i = \sum_{ji \in A} u_{ji} - y_{ji}$, $\beta_i = \sum_{ij \in A} u_{ij} - y_{ij}$ $\rho_i = \min\{\alpha_i, \beta_i\}$

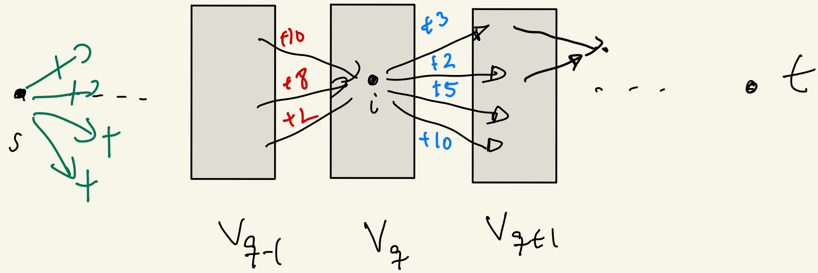




$\rho_s = \sum_{sj \in A} u_{sj} - y_{sj}$, $\rho_t = \sum_{jt \in A} u_{jt} - y_{jt}$ $\rho = \min\{\rho_j \mid j \in [n]\}$

Let i satisfy that $\rho = \rho_i$

(a) Show that we can send an additional g units from i to t and from s to i



Algorithm

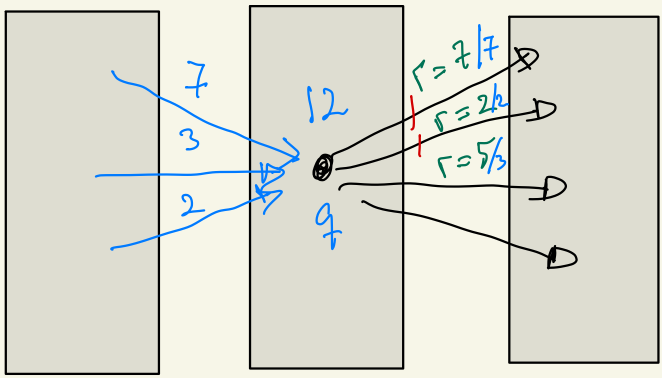
1. set $y \equiv 0$ and calculate g_i $i \in [n]$
   if some $g_i = 0$ goto 6.
2. choose i s.t $g = g_i$
3. push g units from i to t and pull g units from s to i
4. Delete all saturated arcs
   while $\exists j$ s.t j has no arc in or out delete j
5. Calculate new g_i values and g
   If $g_i > 0$ $\forall i$ go to 2.
6. If $g_s = 0$ or $g_t = 0$ then stop
7. While $\exists j \neq s,t$ with $g_j = 0$ delete j
8. Goto 5.

Algorithm

1. set $y \equiv 0$ and calculate ρ_i $i \in [n]$
   if some $\rho_i = 0$ goto 6.
2. choose i s.t $\rho = \rho_i$
3. push ρ units from i to t and pull
   ρ units from s to i
4. Delete all saturated arcs
   while $\exists j$ s.t j has no arc in or out delete j
5. Calculate new ρ_i values and ρ
   If $\rho_i > 0$ $\forall i$ go to 2.
6. If $\rho_s = 0$ or $\rho_t = 0$ then stop
7. While $\exists j \neq s, t$ with $\rho_j = 0$ delete j
8. Goto 5.

(b) The algorithm finds a blocking flow:

Rule: Always push/pull one layer at a time and fill an arc if possible:



(c) show that we can implement the algorithm to run in time $O(n^2)$

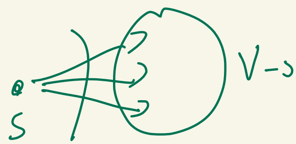
- $O(n)$ push/pull steps
- $O(m)$ work to 'delete' all arcs
- can keep g_i values updated in $O(n \log n)$ use priority Q.

BJG 3.37 Augmenting along maximum capacity augmenting paths

Goal: prove that there will be $O(m \log U)$ augmentations, when $U = \max_{ij \in A} u_{ij}$

Let x^* be a max flow and set $K = |x^*|$

Then $K \leq nU$



Let $x^0 \equiv 0$ so $x^0_{ij} = 0 \;\forall\, ij \in A$

(1) x^* can be decomposed into at most m path flows and some cycle flows

$\Downarrow$

$N(x^0)$ has an (s,t)-path of capacity at least $\frac{K}{m}$

$$X^0 \xrightarrow{P_1} X^1 \xrightarrow{P_2} \cdots \xrightarrow{P_{2m}} X^{2m} \xrightarrow{P_{2m+1}} X^{2m+1}$$

2m augmentations along max cap augmenting paths

<u>Case 1</u> $\exists j \in [2m] : X^j = X^{j+1} = \cdots = X^{2m+1}$

<u>Case 2</u> $|X^{2m+1}| > |X^{2m}|$

Then $\exists\, j \in [2m]$ such that the capacity of P_j is at most $\frac{K}{2m}$

$$(\square) \Rightarrow K' = K - |X^{j-1}| \leq \frac{K}{2}$$

$X^* = X^{j-1} \oplus \tilde{X}$

$\tilde{X}$ flow in $N(X^{j-1})$

max

Repeat same argument for

$y^0 \equiv 0$ in $N(X^{j-1})$ ← max flow value $k' \leq \frac{K}{2}$

$$y^0 \xrightarrow{Q_1} y^1 \xrightarrow{Q_2} \cdots \xrightarrow{Q_{2m}} y^{2m}$$

↖ not maxflow in $N(X^{j-1})$

- either finish in $\leq 2m$ augmentations
- or y^{2m} is not max flow in $N(X^{j-1})$

$\Downarrow$ as we saw before $\exists$ index q s.t.

- capacity of Q_q is at most $\frac{k'}{2m}$

$$k' - |X'| \leq \frac{k'}{2} \quad \text{when } X' = X^{j-1} \oplus y^{q-1}$$

$$\bullet \xrightarrow{2m} \circ \xrightarrow{2m} \bullet \cdots \circ \longrightarrow \circ$$

$X \equiv 0$ $\qquad\qquad$ 0

$K \qquad K/2 \qquad K/4$

$\wedge\!|$

nU at most $\log_2(nU)$ phases each with $O(m)$ augmentations so at most $O(m \log_2 nU)$

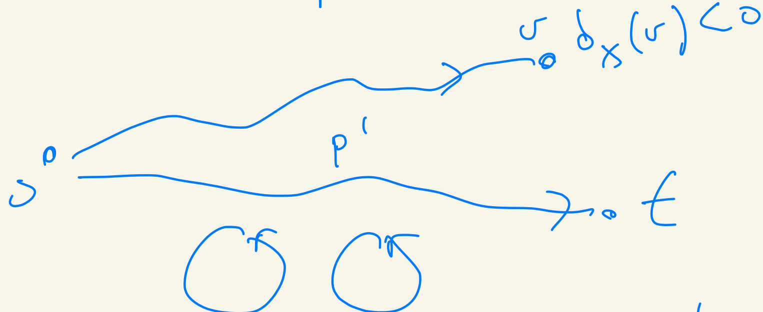
$$= O(m \log_2 n + m \log_2 U)$$

BJG 3.38 $\quad N = (V \cup \{s,t\}, A, l \equiv 0, u)$

Definition A preflow x in N is **maximum** if $-b_x(t) = |x^*|$ when x^* is a maximum (s,t)-flow in N

(a) Let y be a maximum preflow in N show that $\exists\, x$ maximum (s,t)-flow in N s.t $x_{ij} \le y_{ij} \ \forall ij \in A$

y can be decomposed into paths



just keep y on paths of type p'

(5) How fast can we construct x given y?

- decompose y into path and cycle flows $O(nm)$

- set x to be flow whose decomposition is those paths above which an (s,t)-path

$$\underbrace{P_1, P_2, \ldots P_\alpha}_{\text{end in } t}, \underbrace{P_{\alpha+1} \ldots P_\beta}_{\text{don't end in } t}$$

$C_1, C_2 \ldots C_r$

Ahuja 6.33 $N = (V \cup \{s,t\}, A, \ell \equiv 0, u)$

x is an even flow if $x_{ij} \equiv 0 \mod 2 \ \forall ij \in A$

x is an odd flow if $x_{ij} \equiv 1 \mod 2 \ \forall ij \in A$

(a) Claim $u_{ij} \equiv 0 \mod 2 \ \forall ij \in A$

$\Downarrow$ $\exists$ even maximum flow

Let $N_{\frac{1}{2}} = (V \cup \{s,t\}, A, \ell \equiv 0, u_{\frac{1}{2}} \equiv \frac{u}{2})$

and let y be maximum flow in $N_{1/2}$

Then $|y| = u_{\frac{1}{2}}(S, \bar{S})$ for some cut

Set $x_{ij} = 2y_{ij} \ \forall ij$ then x_{ij} is feasible in N

and $|x| = 2|y| = 2u_{\frac{1}{2}}(S, \bar{S}) = u(S, \bar{S})$

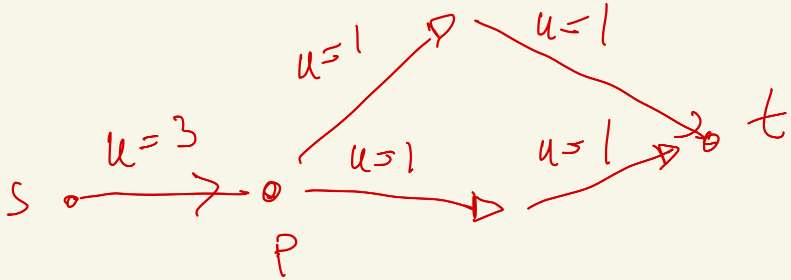
So x is a maxflow and x is even.

(a) is TRUE

(b) claim $u_{ij} \equiv 1 \mod 2 \;\; \forall ij \in A$

$\Downarrow$

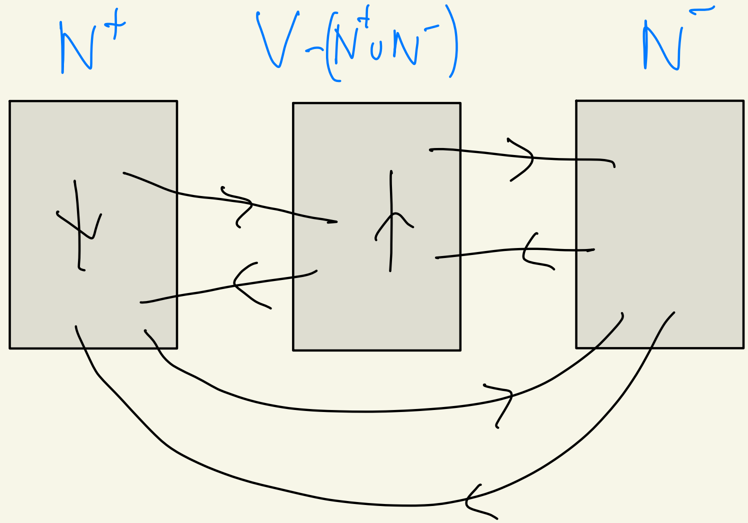
$\exists$ odd maximum flow in N



every maxflow x has $x_{sp} = 2$

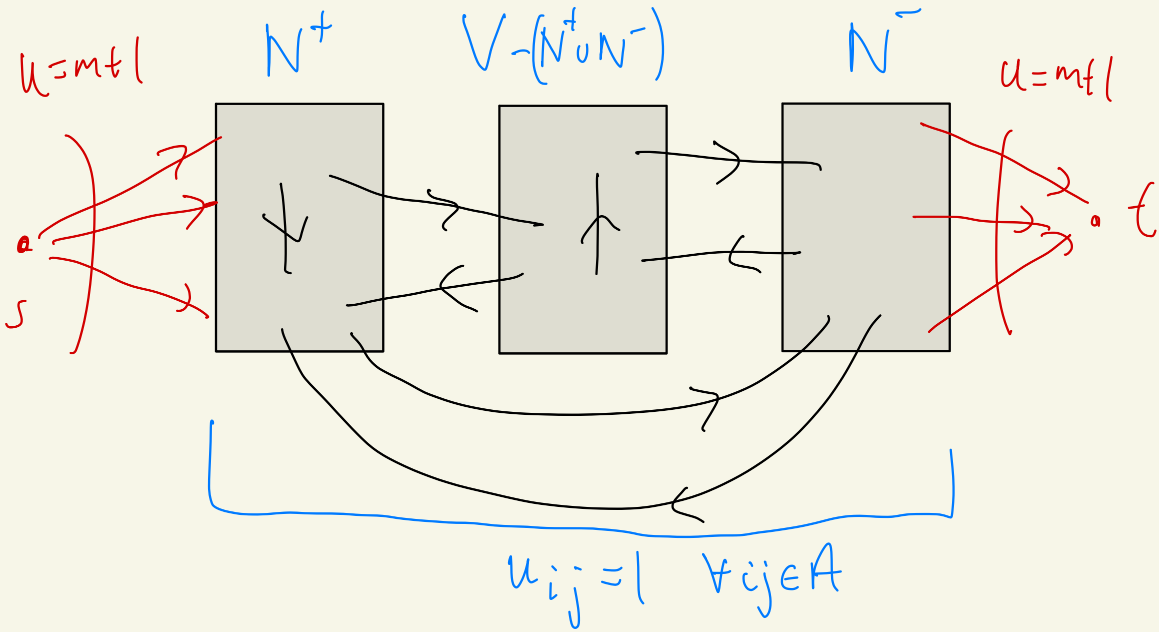
so FALSE

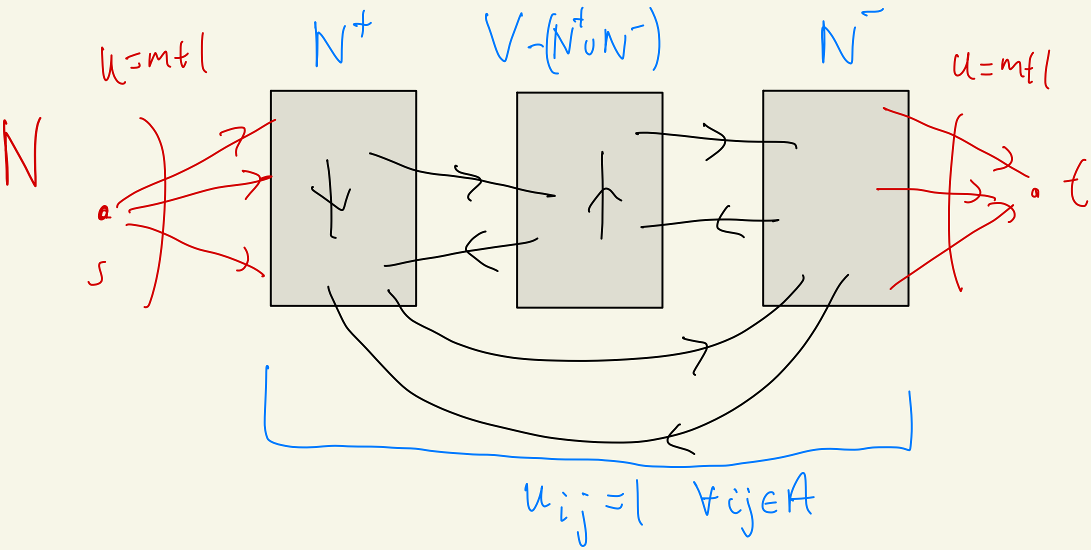
Ahuja 6.45



$m = |A|$

N:





Claim

D has arc-disj. (N^+,N^-)-paths $P_1, P_2, \dots, P_k$

$\Updownarrow$

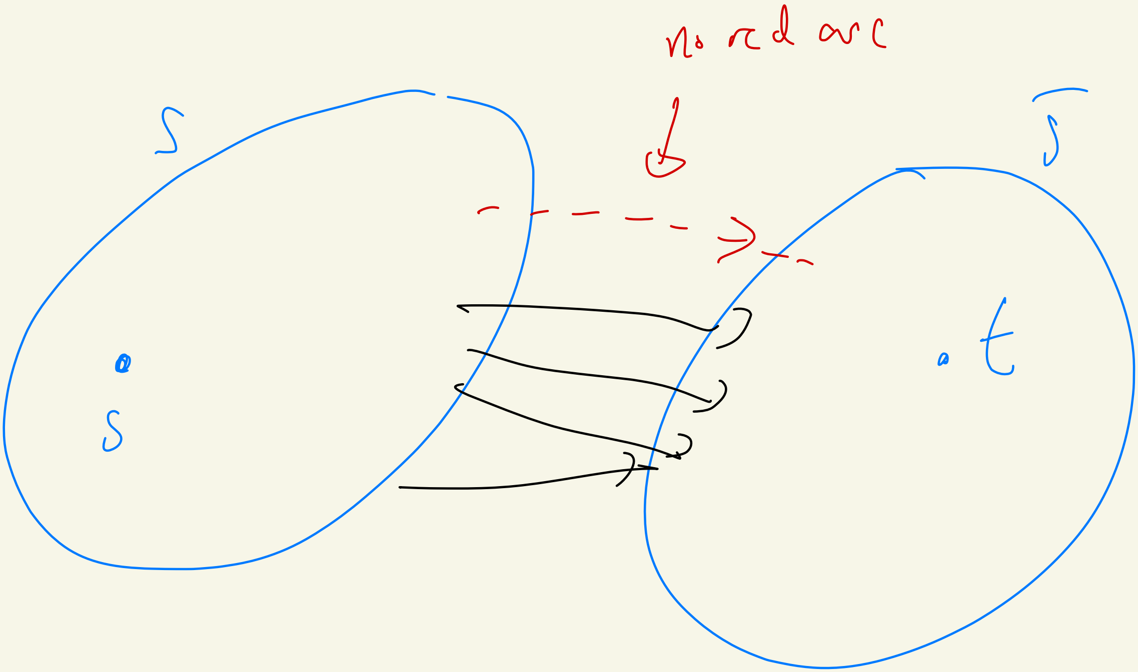
There is an (s,t)-flow of value k in N

$$\max \#\text{arc-disjoint } (N^+,N^-)\text{-paths}$$

$$= \max\{|x| \mid x \ (s,t)\text{-flow in } N\}$$

$$= \min\{u(S,\bar{S}) \mid s \in S,\ t \in \bar{S}\}$$

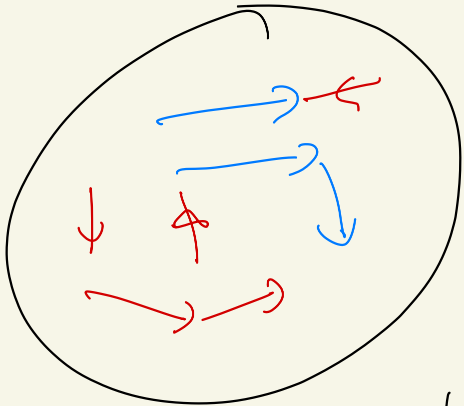
$\overset{?}{=}$ min no of arcs in D that cover all (N^+,N^-)-paths



$(S, \bar{S})$ min cut

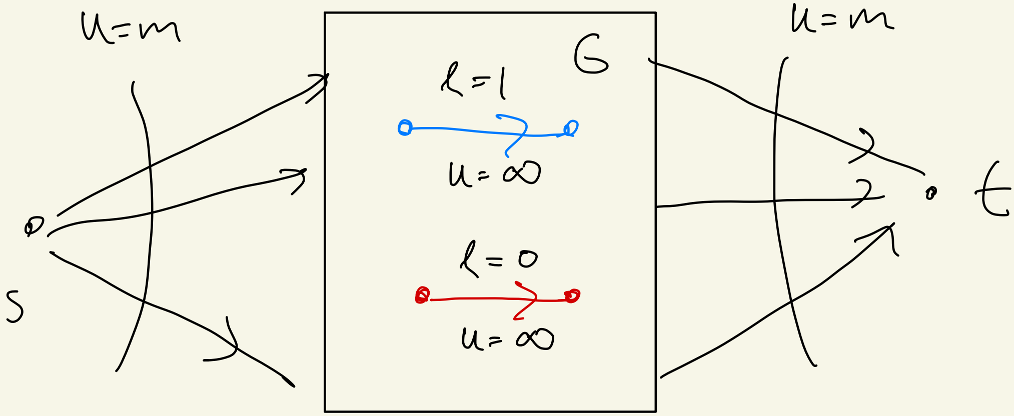
Ahuja 6.47

G acyclic
red and blue arcs

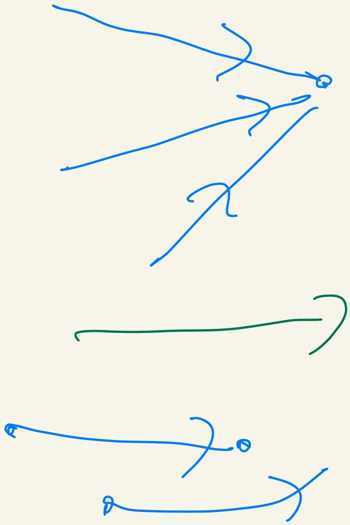


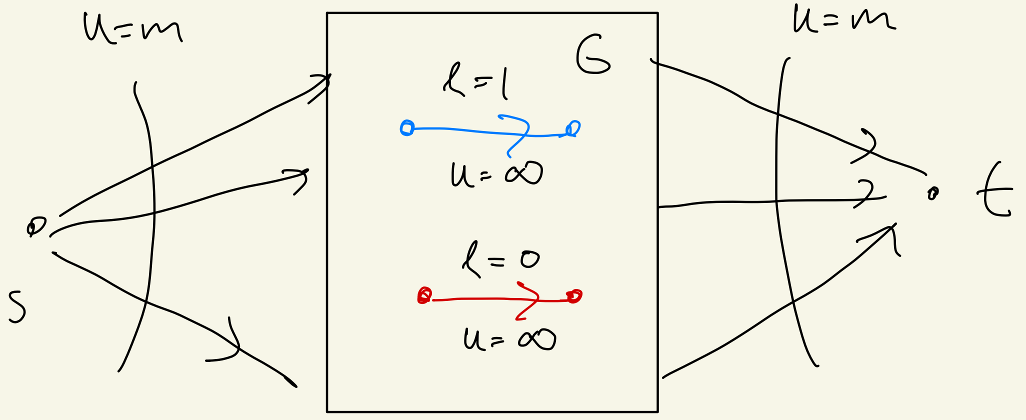
Goal: cover all blue arcs by paths that may use any colour and start anywhere

Claim: min # of paths = max # of blue arcs no two of which can belong to the same path



N_G

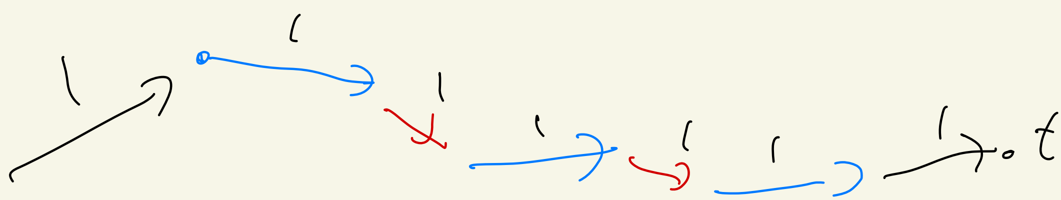




N_G

<u>claim</u> min # paths needed to cover all the arcs in G

= min value of (s,t)-flow in N_G

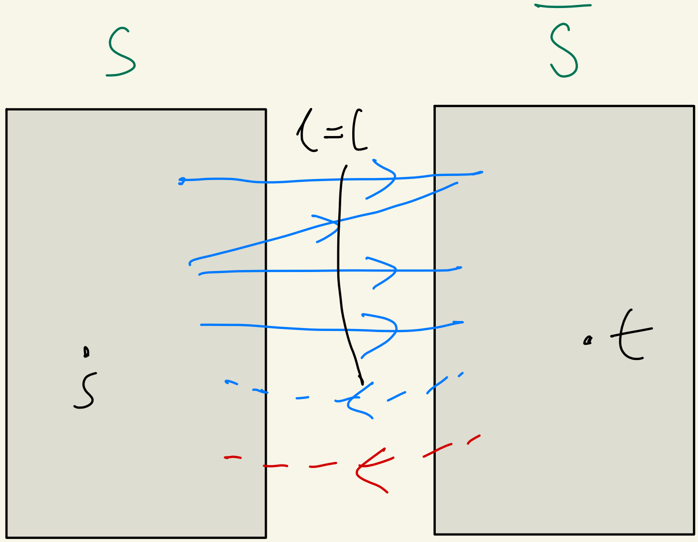


From BJG section 3.9

$\min\{|x| \mid x \text{ is an } (s,t)\text{-flow}\} = \max \gamma(S,\bar{S})$

when $\gamma(S,\bar{S}) = \ell(S,\bar{S}) - u(\bar{S},S)$

If $\gamma(S,\bar{S}) > 0$:

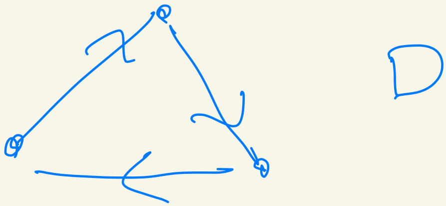


$\gamma(S,\bar{S})$ = # blue arcs from $S \to \bar{S}$

no two of them can belong to the same path!

True for all digraphs?

No



- we need 2 paths to cover blue arcs

- max # blue arcs when no two arcs belong to the same path is 1