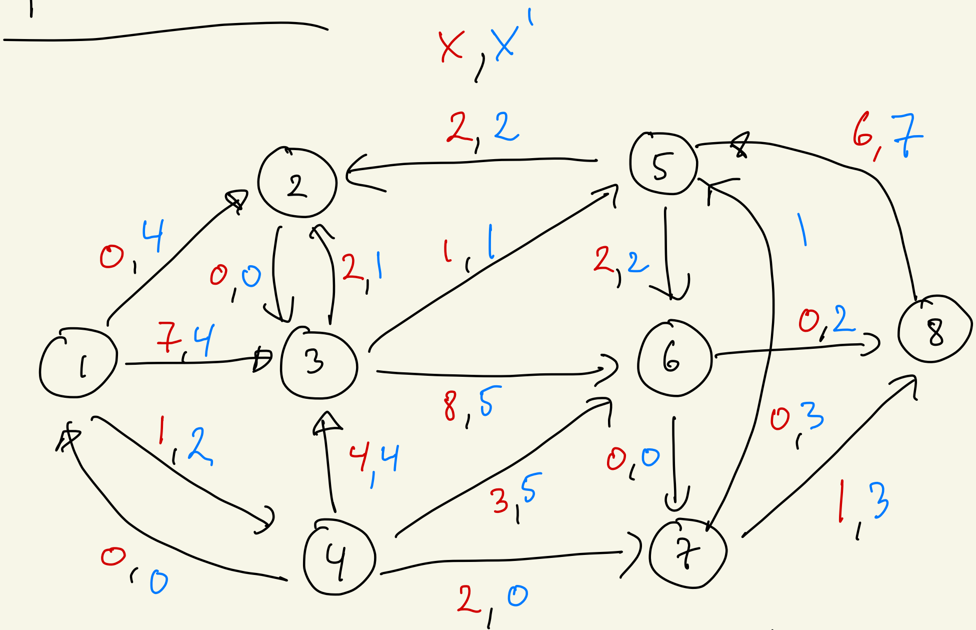
# Problem 1



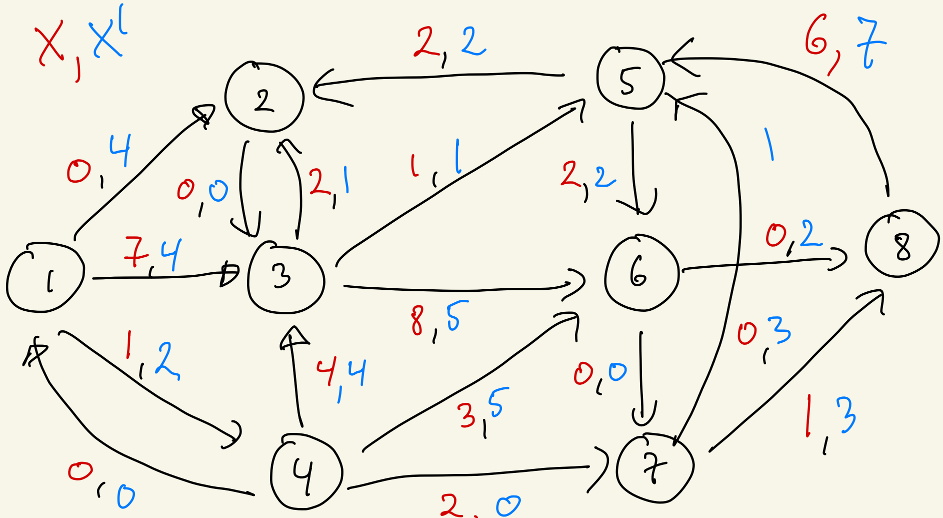

| $v$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $b_x$ | 8 | -4 | 0 | 8 | -3 | -13 | -1 | 5 |
| $b_{x'}$ | 10 | -7 | -1 | 7 | -7 | -10 | 6 | 2 |

To obtain $\bar{x}$ we follow BJG p 107 bottom

if $x_{ij} > x'_{ij}$ then $\bar{x}_{ji} \leftarrow x_{ij} - x'_{ij} + x'_{ji}$

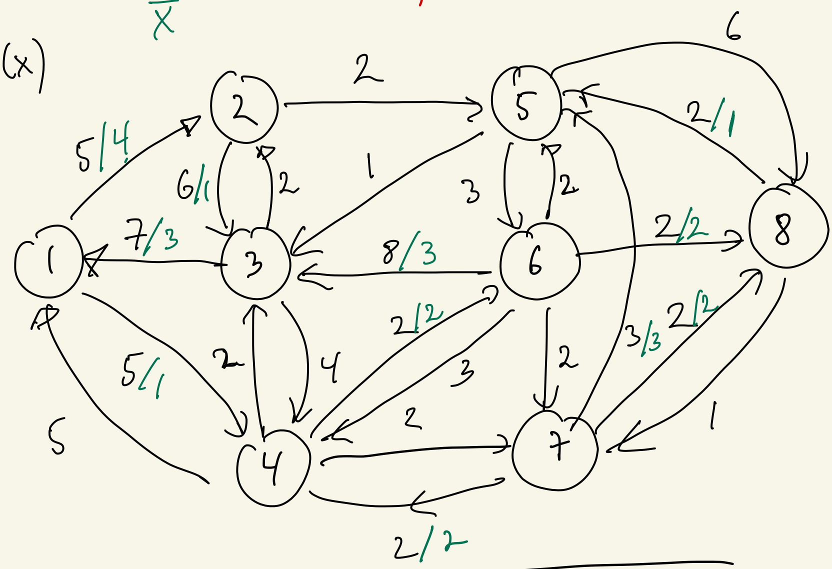
if $x_{ij} < x'_{ij}$ then $\bar{x}_{ij} \leftarrow x'_{ij} - x_{ij} + x_{ji}$

$X, X'$




$\bar{X}$

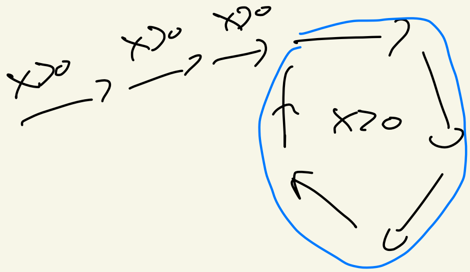
$N(x)$



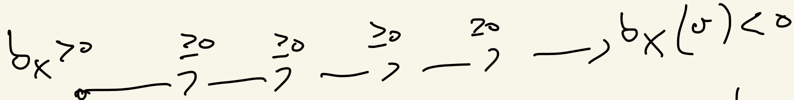

| $v$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $b_X$ | 8 | -4 | 0 | 8 | -3 | -13 | -1 | 5 |
| $b_{X'}$ | 10 | -7 | -1 | 7 | -7 | -10 | 6 | 2 |
| $b_{\bar{X}}$ | 2 | -3 | -1 | -1 | -4 | 3 | 7 | -3 |

b) We decompose $x$ by picking a vertex with $b_x > 0$ and starting a walk then along arcs with $x_{ij} > 0$

This will either stops when a vertex $v$ is repeated and we can extract a cycle flow



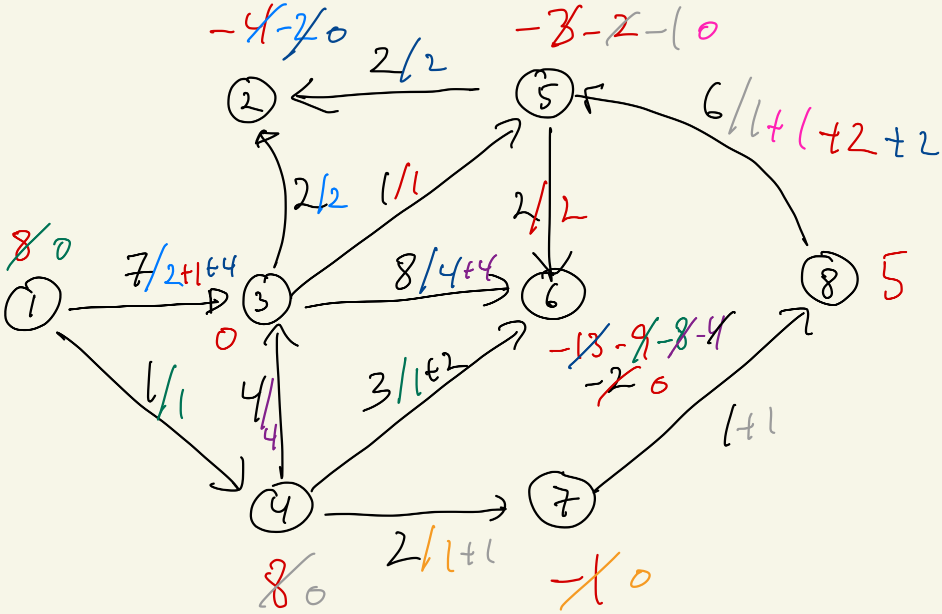

or we reach a vertex $v$ with $b_x(v) < 0$ and then we extract a path flow:




update $x$ in both cases and continue with $x$ being the new flow

When $b_x \equiv 0$ we are done if $x \equiv 0$

else $\exists\, ij$ s.t. $x_{ij} > 0$ and we can start a walk as above and this will always yield a cycle flow to extract



1 → 3 → 2 $\quad \delta = 2$

1 → 3 → 5 $\quad \delta = 1$

1 → 3 → 6 $\quad \delta = 4$

1 → 4 → 6 $\quad \delta = 1$

4 → 3 → 6 $\quad \delta = 4$

4 → 6 $\quad \delta = 2$

4 → 7 $\quad \delta = 1$

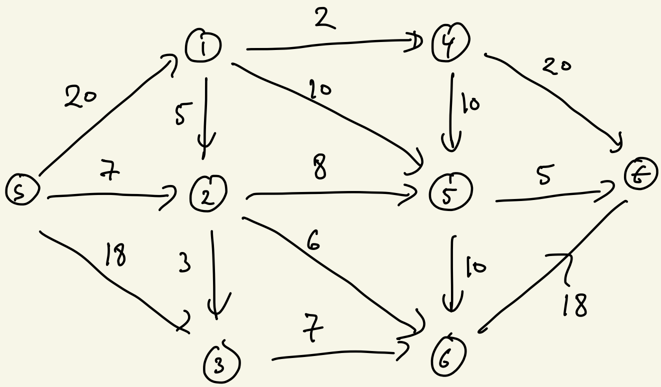
4 → 7 → 8 → 5 $\quad \delta = 1$

8 → 5 $\quad \delta = 1$

8 → 5 → 6 $\quad \delta = 2$

8 → 5 → 2 $\quad \delta = 2$

# Problem 2




a)

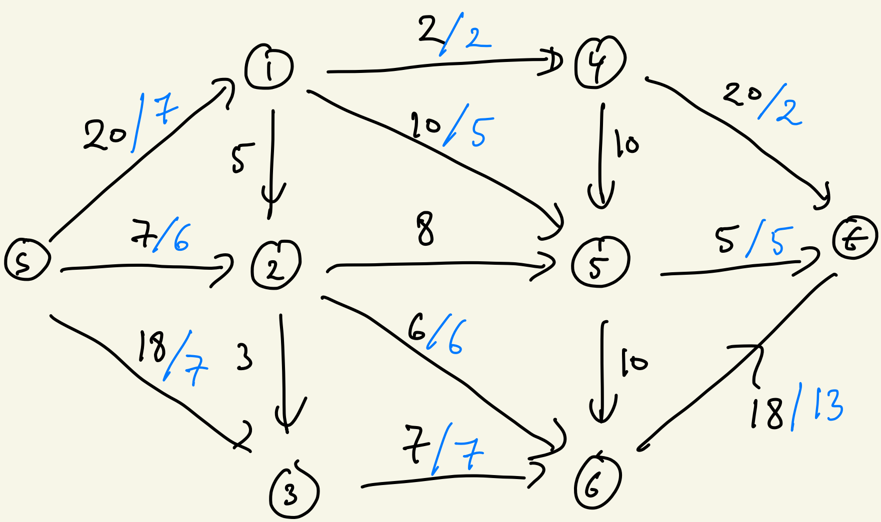
augmenting along

$s \to 1 \to 4 \to t \quad \delta = 2$

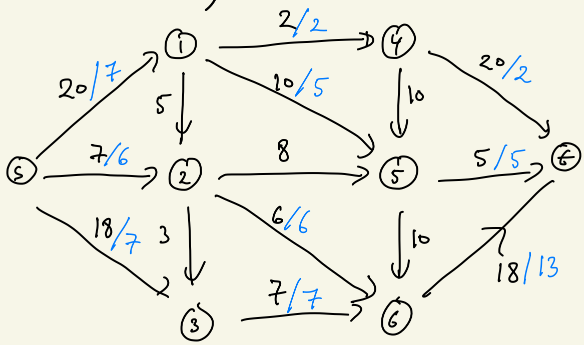
$s \to 1 \to 5 \to t \quad \delta = 5$

$s \to 2 \to 6 \to t \quad \delta = 6$

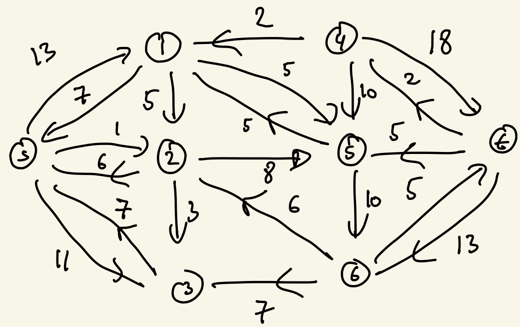
$s \to 3 \to 6 \to t \quad \delta = 7$



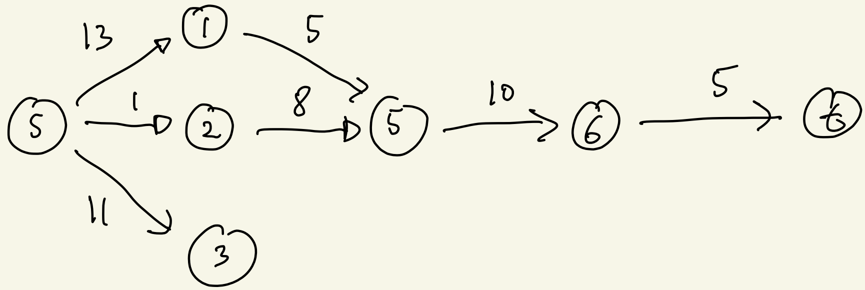
$N, x$




$N(x)$

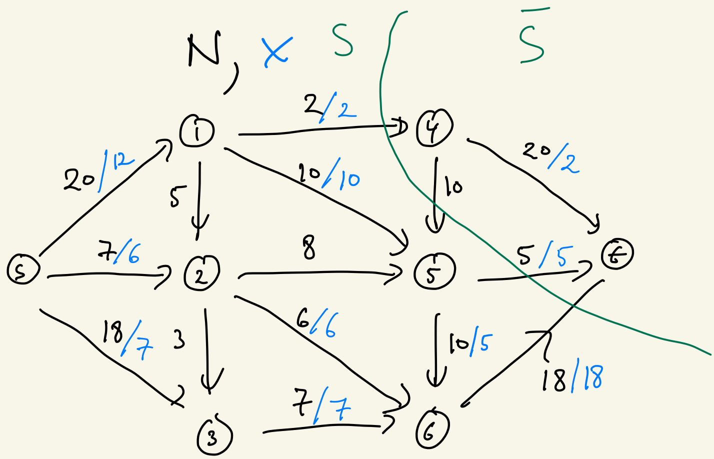



$\mathcal{L}N(x)$




$s \to 1 \to 5 \to 6 \to t \qquad \delta = 5$

$N, x$




$x$ is a maxflow and has value 25

5) $h : V \to \mathbb{Z}$ is a height function with respect to the flow $x$ in $N$ when $h(t) = 0$, $h(s) = n$ and

$h(p) \leq h(q) + 1 \quad \forall pq \in A(N(x))$

<u>operations</u>

lift($v$) and push($pq$)

<u>lift($v$)</u> may be applied precisely when

1. $b_x(v) < 0$
2. $h(v) \leq h(w) \quad \forall$ arcs $vw$ in $N(x)$

result $h(v) \leftarrow \min\{h(w) \mid vw \in A(N(x))\} + 1$

<u>push $pq$</u> may be applied when

1. $b_x(p) < 0$
2. $h(p) = h(q) + 1$

result: $x_{pq} \leftarrow x_{pq} + \min\{-b_x(p), r_{pq}\}$

# PFP alg (sketch)

1. set $x_{ij} = 0 \quad \forall ij \in A$

3 initialize $h$ via distances to $t$ in $N(x)$, $h(s) \leftarrow n$

2. $\forall sq \in A$: $x_{sq} \leftarrow u_{sq}$

4. while $\exists v \notin \{s,t\}$: $b_x(v) < 0$

   select arbitrary such $v$

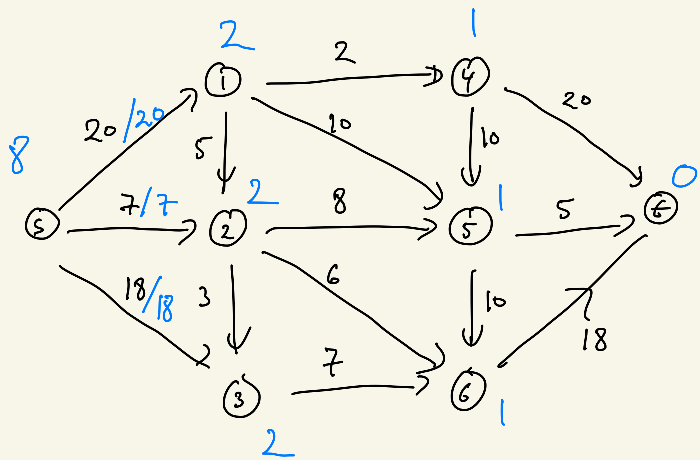
   if we can apply a push from $v$

   do it

   else lift($v$)

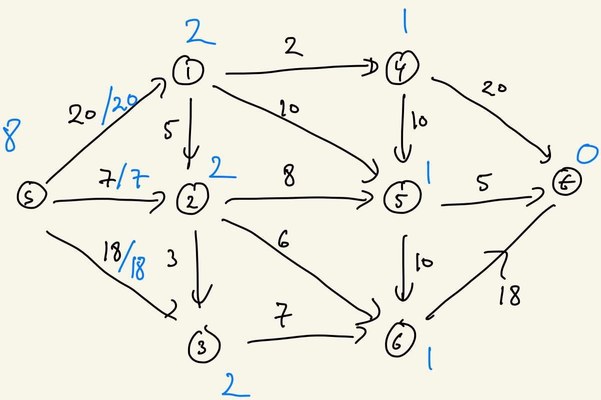
5. return $x$.

c)

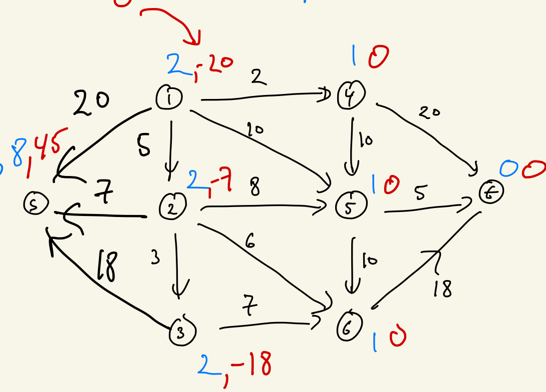



initial $h$ and $x$

X




N(x)




select vertex 1   $b_x(1) = -20$

push $1 \to 4$   2 units

$1 \to 5$   10 units

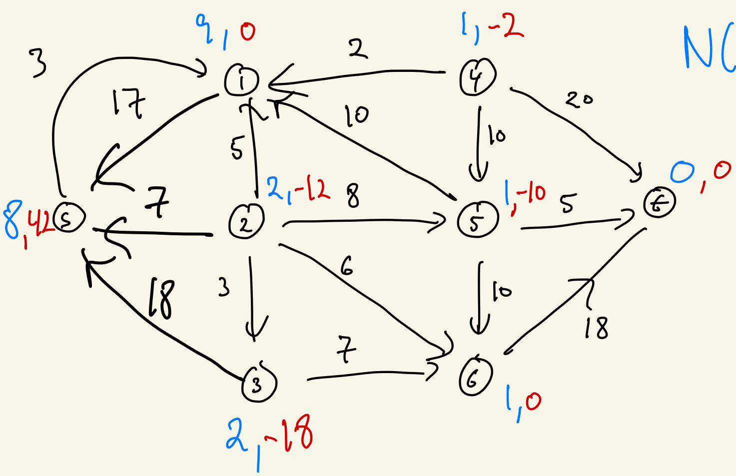
lift(1)   $h(1) \leftarrow 3$

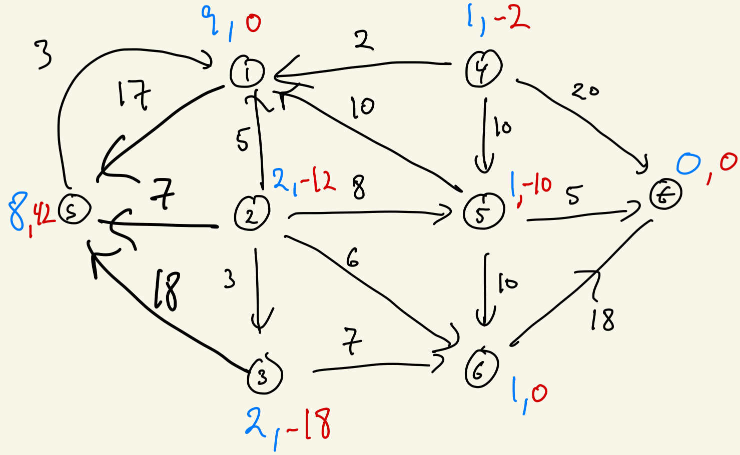
push $1 \to 2$   5 units

lift(1)   $h(1) \leftarrow 9$

push $1 \to 5$   3 units

N(x)

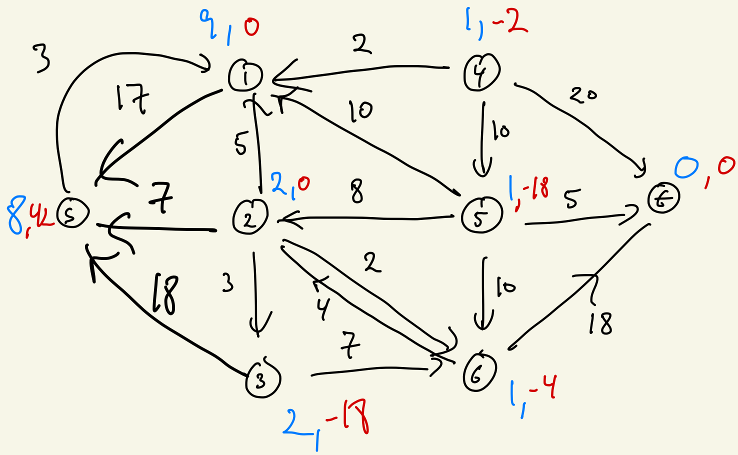




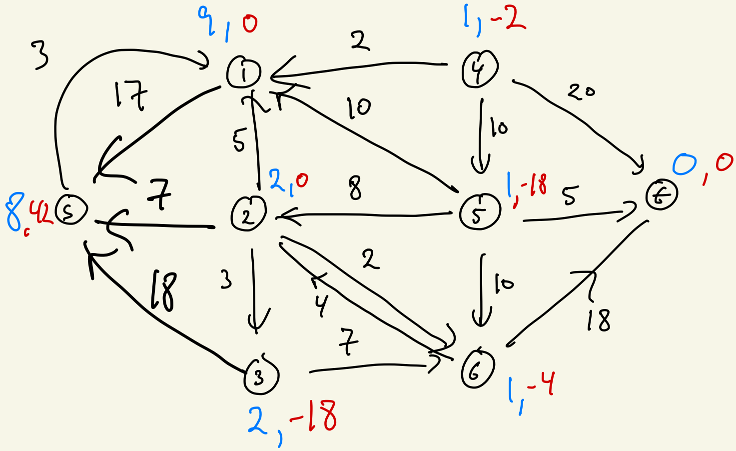
Select vertex 2:

push $2 \to 5$ 8 units

push $2 \to 6$ 4 units ($b_x(2) = -4$ after first push)



$N(x)$

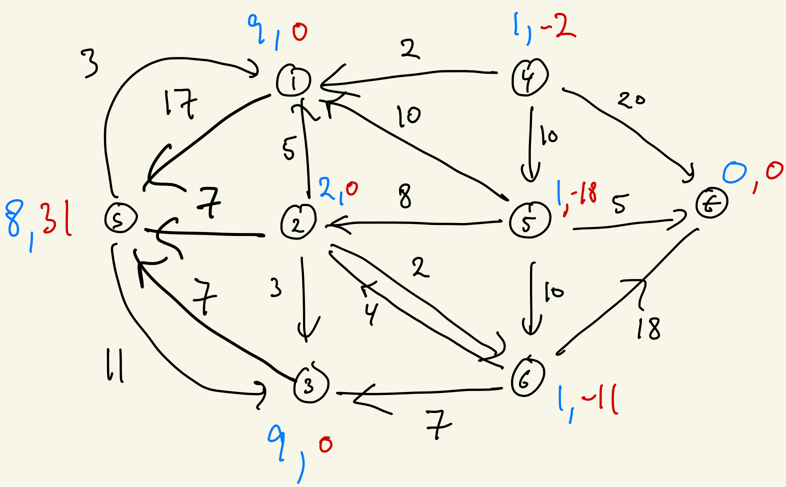



Select vertex 3

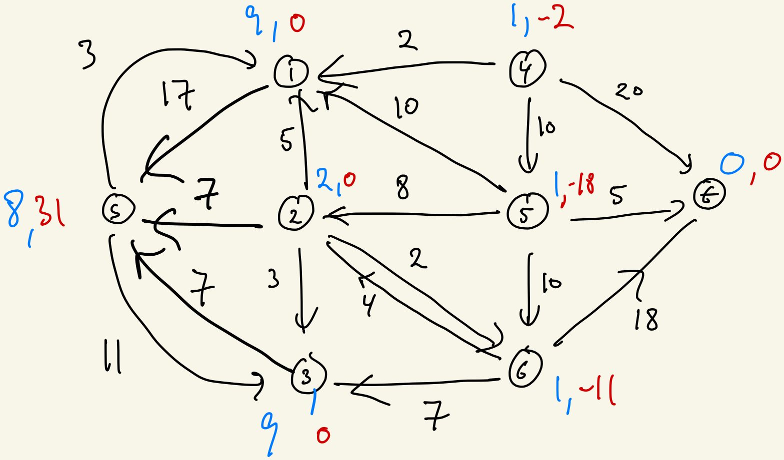
push 3 → 6 7 units

lift(3) h(3) ← 9

push 3 → s 11 units




N(x)



select vertex 4:

push $4 \to t$ 2 units

select vertex 5:

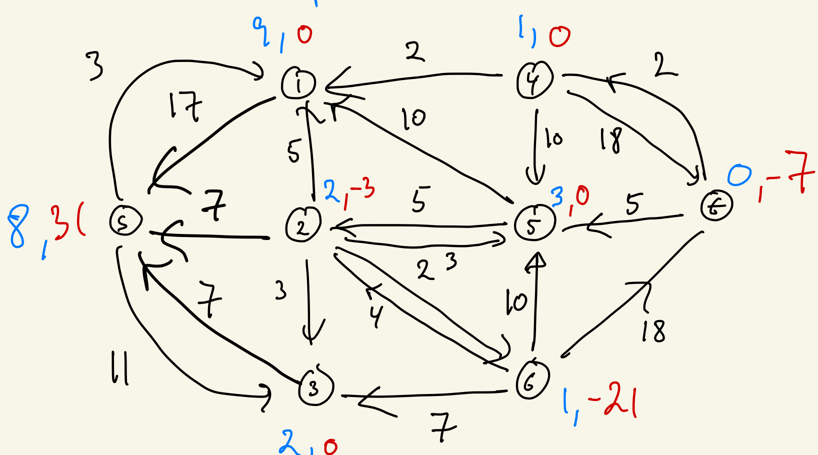
push $5 \to t$ 5 units

lift(5) $h(5) \leftarrow 2$

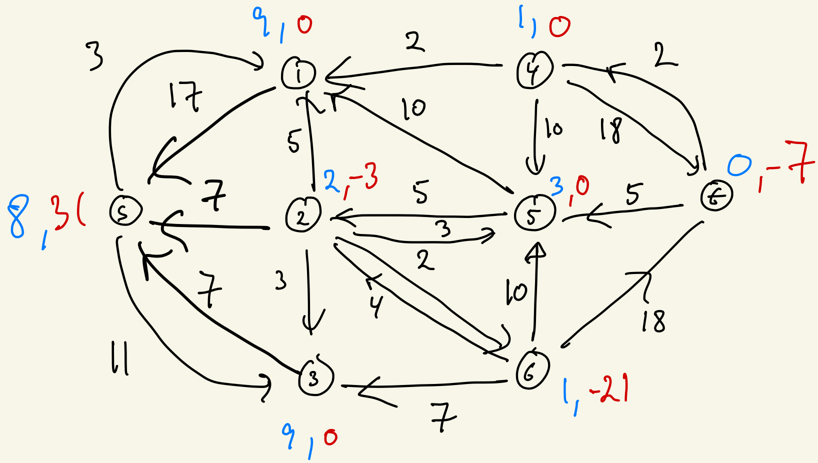
push $5 \to 6$ 10 units

lift(5) $h(5) \leftarrow 3$

push $5 \to 2$ 3 units



$N(x)$

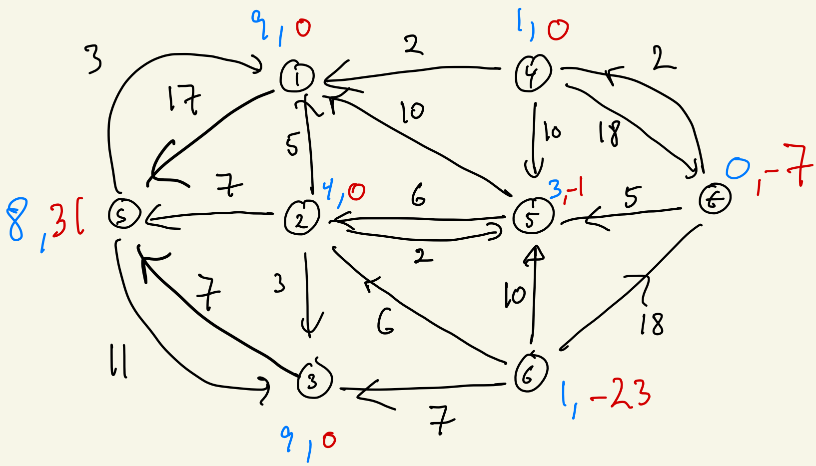



Select vertex 2

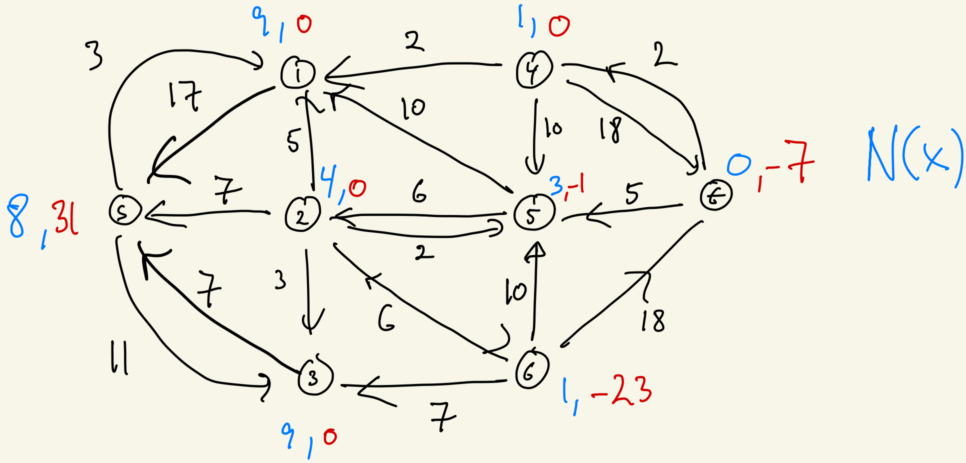
push 2 → 6   2 units

lift(2)   h(2) ← 4

push 2 → 5   1 unit




N(x)

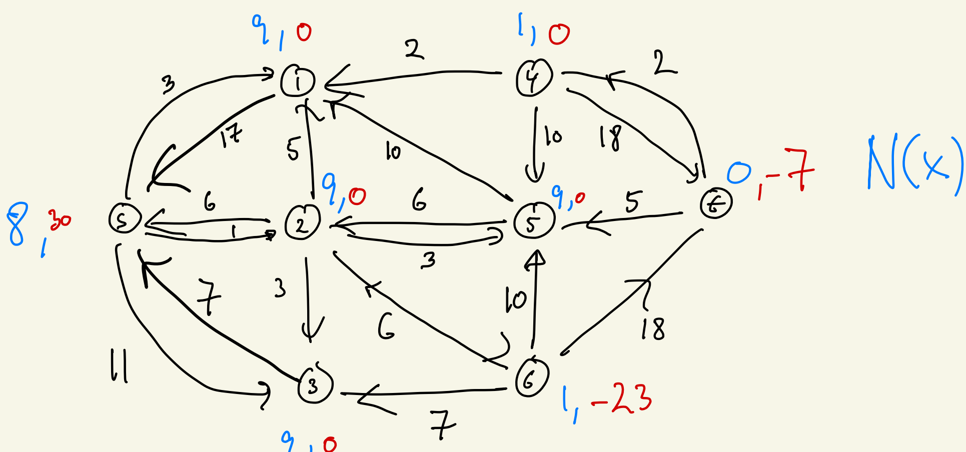


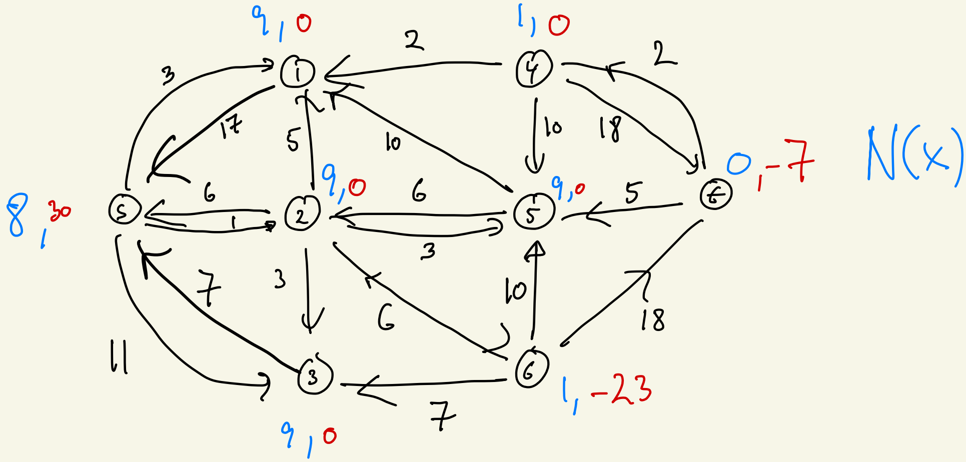

Select vertex 5 and 2 alternatively

$h(5) \leftarrow 5$ push $5 \xrightarrow{1} 2$, $h(2) \leftarrow 6$ push $2 \xrightarrow{1} 5$, $h(5) \leftarrow 7$ push $5 \xrightarrow{1} 2$

$h(2) \leftarrow 8$ push $2 \xrightarrow{1} 5$, $h(5) \leftarrow 9$ push $5 \xrightarrow{1} 2$.

Finally lift(2) $h(2) \leftarrow 9$ push $2 \xrightarrow{1} s$





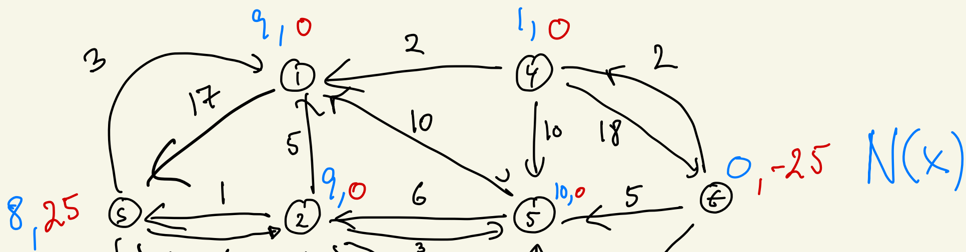

select vertex 6:

push $6 \to t$ 18 units

lift 6 $h(6) \leftarrow 10$

push $6 \to 2$ 5 units

push $2 \to s$ 5 units




no more active vertices

$\Downarrow$

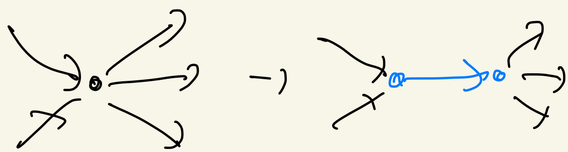
alg terminates

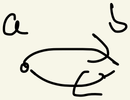
# Problem 3

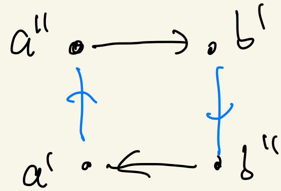
- We are looking for vertex disjoint paths from the vertices $Y = \{v_{i_1 j_1}, \dots, v_{i_r j_r}\}$ to $V'$

a)

- Given $G_{q,q}$ and $Y$ we first make a digraph $D_{q,q}$ by replacing each edge by a directed 2-cycle 

- Let $D'_{q,q}$ be obtained from $D_{q,q}$ by vertex splitting 

then each 

becomes 

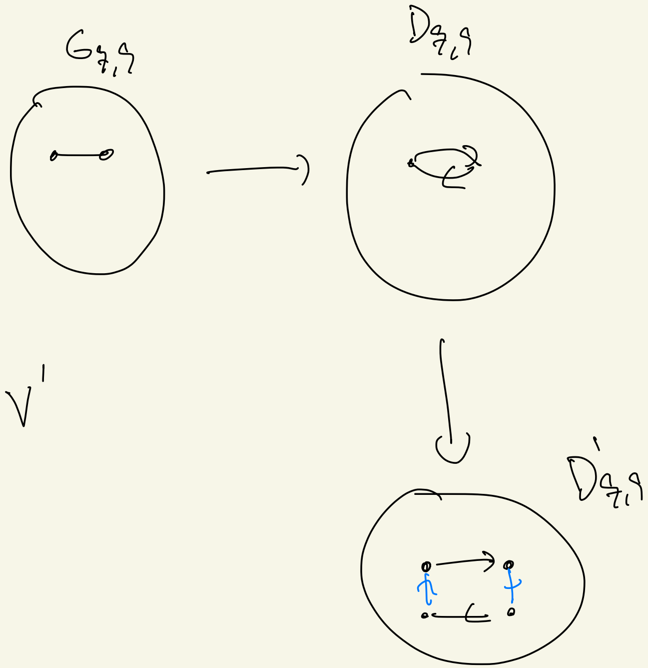
obtain $N$ from $D'_{q,q}$ as follows

- add new vertices $s, t$
- add arcs $s \to Y_1' \cup Y_2'$ $\;Y_2'' \cup Z'' \to t$
  when $Y_1 = Y \setminus V'$, $Y_2 = Y \cap V'$, $Z = V' \setminus Y$
- set all capacities to 1
- set all lowerbound to 0

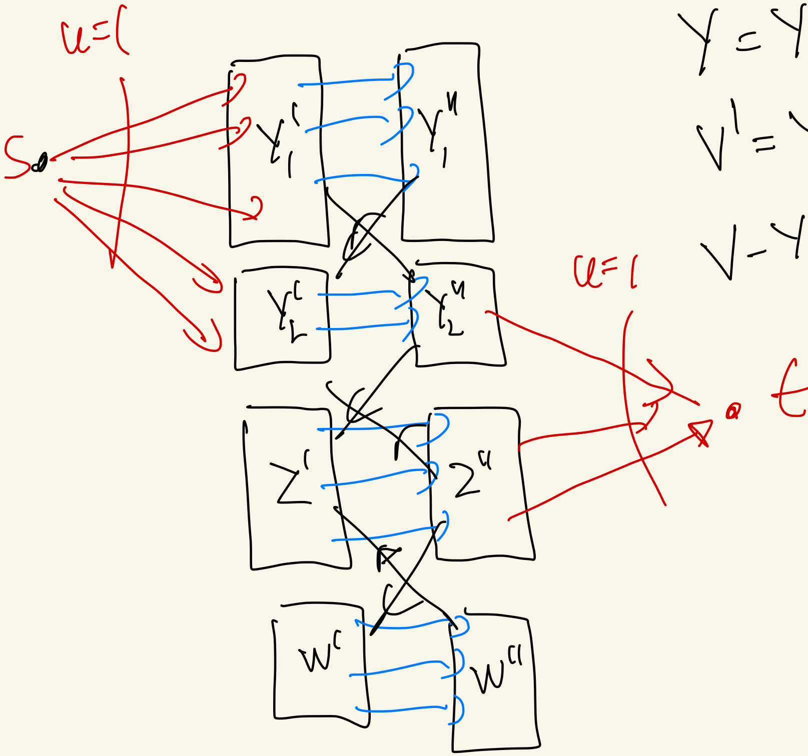
Claim $N$ has an $(s,t)$-flow of value $r = |Y|$
$\Updownarrow$ The escape problem $\langle G_{q,q}, Y \rangle$
has a solution

$\Downarrow$: let $x$ be an integer flow of value $r$ in $N$
Decompose $x$ into $r$ path flows of value 1
along $P_1', P_2' \ldots P_r'$ and some cycles

- The paths are vertex disjoint as $u = 1$
  on the arcs $(a' \to a'')$
- let $Q_i$ be obtained from $P_i'$ by deleting
  $s$ and $t$ and contracting all the arcs




$Z = V'$




$Y = Y_1 \cup Y_2$

$V' = Y_2 \cup Z$

$V - Y - V' = W$

- $Q_1, Q_2 \ldots Q_r$ are disjoint paths in $G_{\mathcal{F},\mathcal{F}}$ from $Y$ to $V'$ so they form a solution

$\Uparrow$ - let $R_1, R_2 \ldots R_r$ be disjoint paths from $Y$ to $V'$ in $G_{\mathcal{F},\mathcal{F}}$ (a solution)

- and let $R_1', R_2' \ldots R_r'$ be the corresponding paths in $D'_{\mathcal{F},\mathcal{F}}$

- Then we obtain an $(s,t)$-flow of value $r$ in $N$ by sending one unit of flow along $s \to R_i' \to t \quad \forall i = 1, 2, \ldots, r$

## Complexity:

N is a simple unit capacity network with $n = 2q^2+2$ vertices and

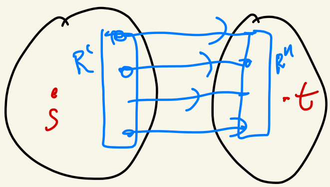
$4q(q-1)+|Y|+4(q-1)$ arcs which is $O(q^2)$

So Dinic solves the problem in time

$$O(n^{1/2}m) = O(q \cdot q^2) = O(q^3)$$

## Certificate

If we change the capacity of all original arcs of $D'_{q,q}$ to $\infty$ then max flow value is unchanged and now every min cut is of the form



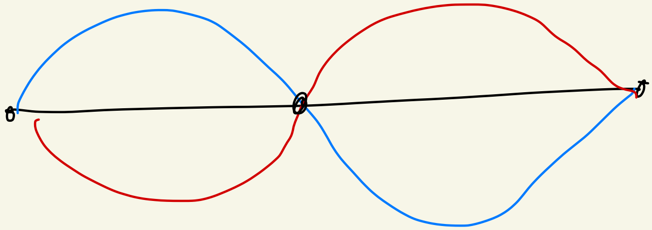

with $|R| < |Y| = r$

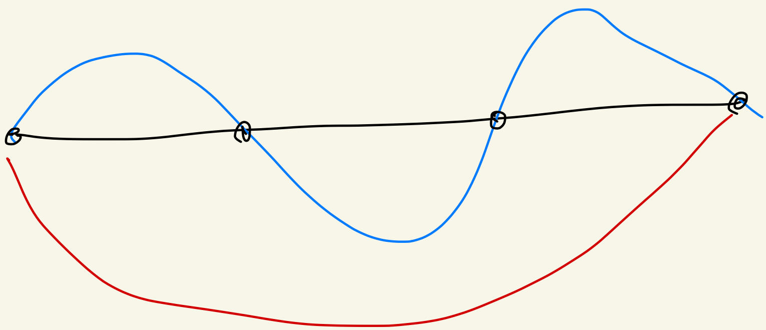
So every path from $Y$ to $V'$ contains a vertex from $R$. Hence $R$ is a certificate that there is no set of escape routes

**Note** the change does not affect the running time of Dinic's alg (why?)

b) Now we allow several paths to pass through the same vertex in $G_{\xi,\xi}$ and want to minimize the maximum # of times any vertex is used in a set of paths $P_1, P_2, \dots, P_r$ from $Y$ to $V'$

- Let $k$ be this minimum that is, there is a solution where some vertex is used $k$ times, but every solution uses some vertex at on at least $k$ paths
- To look for a solution that uses no vertex more than $\xi$ times we change $N$ by letting $u_{v'v''} = \xi$ $\forall v \in V(G_{\xi,\xi})$
- By our argument in a) the new $N$ has a flow of value $r$ if and only if $G_{\xi,\xi}$ has path $P_1, \dots P_r$ from $X$ to $V'$ s.t. no vertex is on more than $\xi$ paths
- Find $k$ by binary search



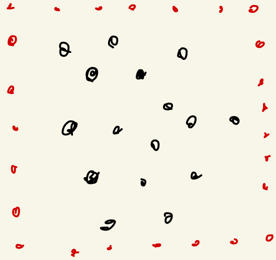


a) $1 \le k \le |Y| \le n$

this takes time $O(\log n)$ time, the time to find a flow of value $|Y|$

If we run Ford Fulkerson this takes time $O(|Y| m)$ per max flow calc.

c)

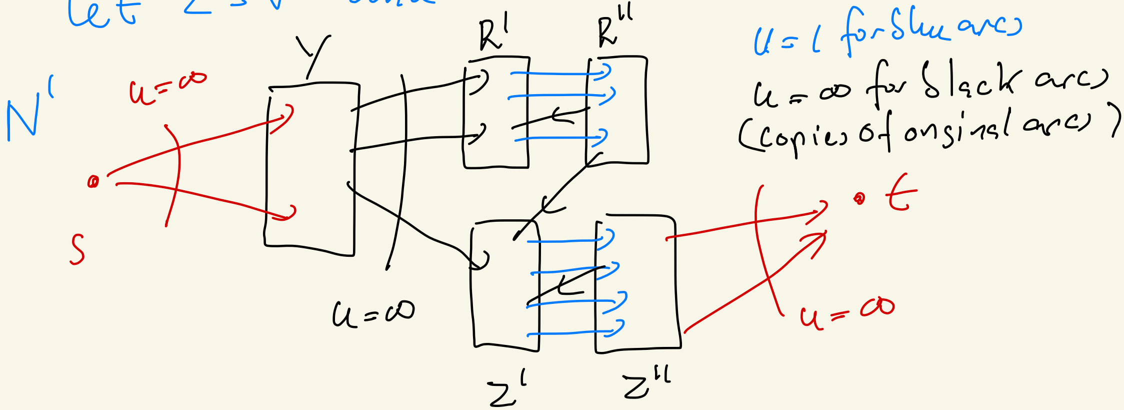


$Y = \circ$ has no vertex in $V'$

$V' = \circ$

let $N'$ be as defined below

let $Z = V'$ and $R = V - (Y \cup Z)$




- Every $Y \to V'$ path $P$ in $G_{q,q}$ corresponds to an $(s,t)$ path $P'$ in $N'$ in which every second arc is blue except for the arcs incident to $s$ and $t$.

- so $W \subseteq V - Y$ intersects all $Y \to V'$ paths in $G_{q,q}$ if and only if $W'$ intersects all $(s,t)$-paths in $t$

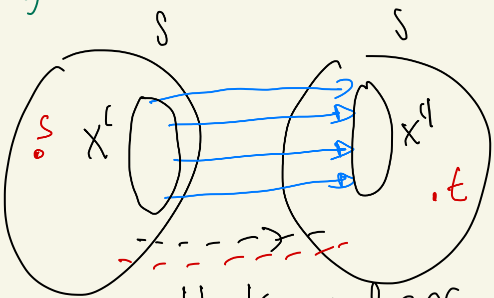
So to show that we can find a minimum X intersecting all (Y, V')-paths via a max flow calculation, it suffices to prove the following.

<u>Claim</u>

$$\max\{|x| \mid x \text{ is an } (s,t)\text{-flow in } N'\}$$

$$= \min\{|X| \mid \text{Every } (Y, V')\text{-path in } G_{q,q} \text{ goes through } X\}$$

- Every min cut in $N'$ is of the form




all

no black or red arc

- By max flow min cut thm:

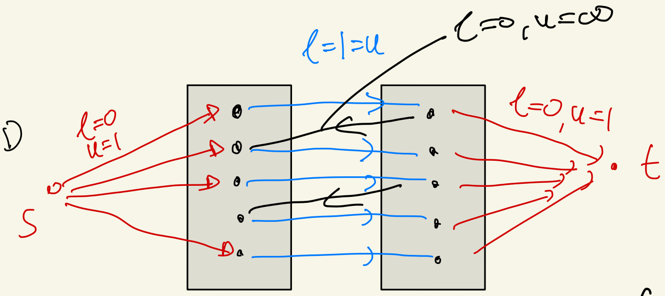
$$|x^*| = u(S, \bar{S})$$
$$= |X'|$$
$$= |X|$$

$\square$

# Problem 4

1. Let $D=(V,A)$ be acyclic and obtain $N_D$ from $D$ by vertex splitting and adding new vertices $s,t$ as follows

$N_D$




**Claim:** $N_D$ has a feasible $(s,t)$-flow of value $k$

$\Updownarrow$

$D$ has a path cover with $k$ paths

$\Downarrow$ An integer flow $x$ of value $k$ decomposes into $k$ path flows of value 1 along paths $P_1 \dots P_k$. $P'_1, P'_2, \dots, P'_k$ is a path cover of $D$, where $P'_i = P_i$ minus $s,t$ and blue arcs contracted

$\Uparrow$ Let $Q_1, Q_2 \dots Q_k$ be a path cover in $D$. Send 1 unit along $sQ'_it$, $i \in [k]$ $\rightarrow$ flow of value $k$ in $N_D$, where $Q'_i$ is obtained from $Q_i$ by inserting the blue arcs

Hence we can find min path cover of $D$ by finding a minimum value flow in $N_D$ and this can be done by 2 max flow calc. see BJG sec 3.9.

2. $N = (V, A, \ell \equiv 0, u \equiv 1, b, c)$

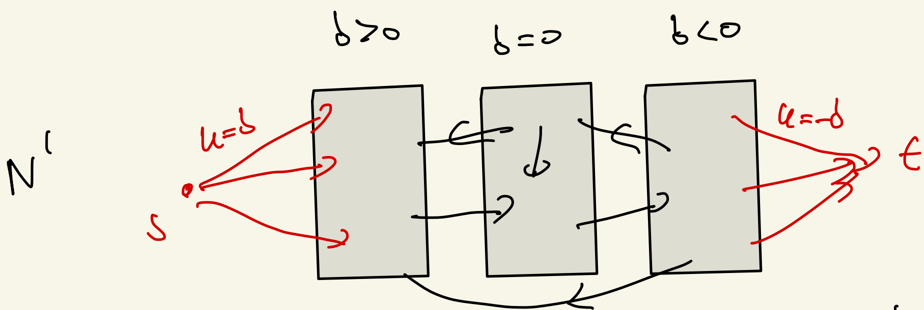
We have seen the the complexity of the buildup algorithm is $O(n^2 m M)$, when $M = \max\{|b(v)| \mid v \in V\}$

Observe that $M \leq n$ must hold if there is a feasible flow as $b_x(i) \leq \sum_{ij \in A} u_{ij} \leq n$

Hence the complexity becomes $O(n^3 m)$

We may either check for a feasible flow first (see 3.) or note that the algorithm can run for at most $n^2$ iterations as the cut $(U_x, V-U_x)$ has capacity less than $|U_x| \cdot |V-U_x| < n^2$

3. $N=(V, A, \ell \equiv 0, u \equiv 1, b)$
we can check the existence of a feasible flow in $N$
by solving a maxflow problem in $N'$:

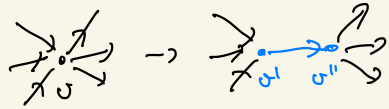



$N'$ is not a unit capacity network but when we calculate distance classes $V_0=s, V_1, \dots, V_{\omega-1}, V_\omega \ni t$ from $s$
the capacity of every cut $(V_0 \cup \dots \cup V_i, V - V_0 \cup \dots \cup V_i)$ is bounded by $|V_i| \cdot |V_{i+1}|$ so as in the proof of lemma 3.7.3 in BJG
we see that $dist_N(s,t)$ is roughly $\frac{2n}{\sqrt{|x^*|}}$ when $x^*$ is a max flow in $N$.
Using this as in the proof of Thm 3.7.4, we get the desired complexity.

4. $D=(V,A)$ has a cycle factor

$\Updownarrow$

$D'=(V',A')$ has a cycle factor, where

$D'$ is obtained from $D$ by vertex splitting 


Let $N=(V',A',l,u)$ where $l(v'v'')=1$ $\forall v\in V$ and $l=0$ for all other arcs, $u(v'v'')=1$ $\forall v\in V$ and $u=\infty$ otherwise

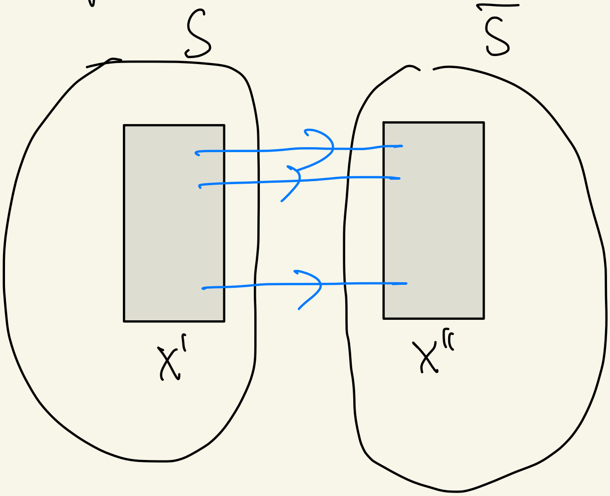
Claim: $N$ has a feasible circulation $\Leftrightarrow$ $D$ has a cycle factor

$\Leftarrow$: Let $C_1, C_2 .. C_r$ be a cycle factor of $D$ and let $C_1', C_2' .. C_r'$ be the corresponding cycles in $N$. Then we obtain a feasible circulation $x$ in $N$ by sending one unit of flow along each $C_i'$

$\Rightarrow$) Let $x$ be a feasible circulation in $N$. $x$ decomposes into cycle flows of value 1 along vertex disjoint cycles $W_1', W_2', .. W_p'$ (as $u_{v'v''}=1$ $\forall v\in V$)

Let $C_j$ be obtained from $W_j'$ by contracting the blue arcs, then $C_1, C_2 .. C_p$ is a cycle factor in $D$.

By Hoffman's circulation theorem $N$ has no feasible circulation precisely if $\exists\, S \neq \emptyset, V$ s.t. $\ell(S,\bar{S}) > u(\bar{S},S)$:




let $X \subseteq V$ be such that $\ell(S,\bar{S}) = |X|$

- $X$ is independent as an arc $v \to w$ $v,w \in X$ gives arc $v'' \xrightarrow{u=\infty} w'$ in $N$
- If there are $k = |X|$ internally disjoint paths $P_1 \dots P_k$, from $X$ to $X$ in $D$, then the corresponding paths $P'_1, P'_2, \dots, P'_k$ all cross from $\bar{S}$ to $S$ so

$$u(\bar{S},S) \geq k = |X| = \ell(S,\bar{S}) \ \lightning$$

- So there is no such set of paths in $D$

- This means that $D'$ does not have $k$ internally disj paths from $X''$ to $X'$ so by Mengers theorem we can kill all such paths by removing a set $Z'$ of vertices whose $|Z'| < |X|$

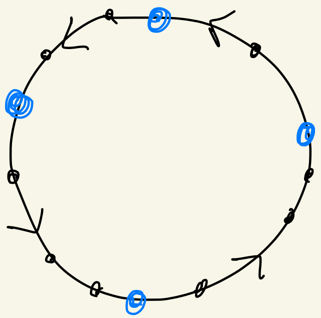
  Back in $D$, the corresponding set $Z$ kills all $X \to X$ paths

- We have shown that

  $D$ has no cycle factor $\Rightarrow$ $Z$ exists

  and if

  $D$ has a cycle factor $\underbrace{}_{C_1, C_2 \dots C_t}$ then $Z$ cannot exist as the cycles provide such paths for every subset $X \subseteq V$

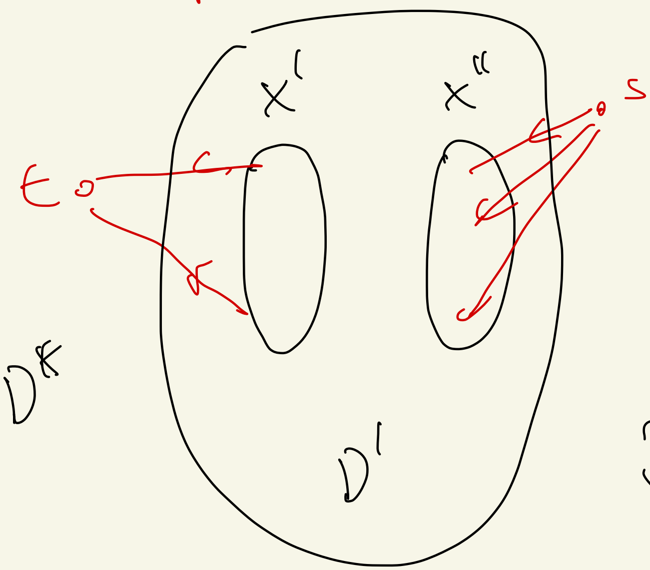


$\circ$ = in $X$

if $C_i$ contains $r$ vertices from $X$

then $C_i$ contains $r$ $X \to X$ paths

Comment of use of Menger




$D'$ has $k = |X|$ disjoint $X'' \to X'$ paths

$\Downarrow$

$D^*$ has $k$ internally disj $(s,t)$-paths

5. $x$ feasible in $N = (V, A, l \equiv 0, u, b, c)$

assume $N(x)$ has a unique negative cycle $W$ and $c(W) = -10$, $\delta(W) = 5$

- $x' = x \oplus \delta(W)$ is feasible and has cost $cx + \delta(W) \cdot c(W) = cx - 50$

- suppose $x'$ is not a min cost flow in $N$ and let $x''$ be a feasible flow with $cx'' < cx'$

- let $\bar{x} \in N(x)$ be such that $x'' = x \oplus \bar{x}$
  Then $\bar{x}$ is a circulation and hence decomposes into cycle flows along cycles $W_1, W_2 \dots W_k$

Now $cx'' = cx + \sum_{i=1}^{k} \delta(W_i) \cdot c(W_i)$

$$< cx' = cx + \delta(W) \cdot c(W)$$

so at least 2 of the cycles $W_1, \dots W_k$ must be negative contradicting that $W$ is only neg cycle in $N(x)$

## Problem 5

Given a set $S = \{b_1, b_2, \dots b_p\}$ of bookings

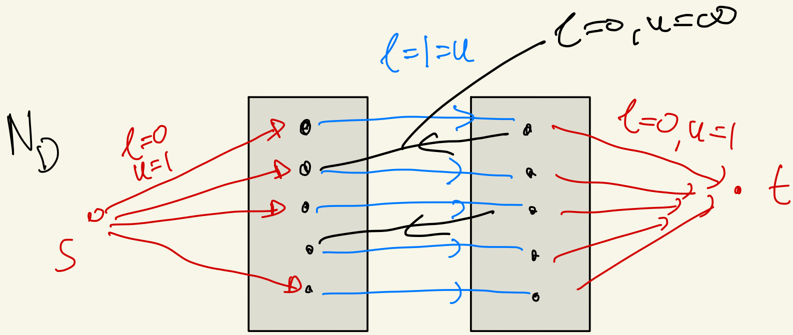
a) form a digraph $D = (V, A)$ where $V = \{v_1, v_2 \dots v_p\}$ and $v_i \to v_j \in A$

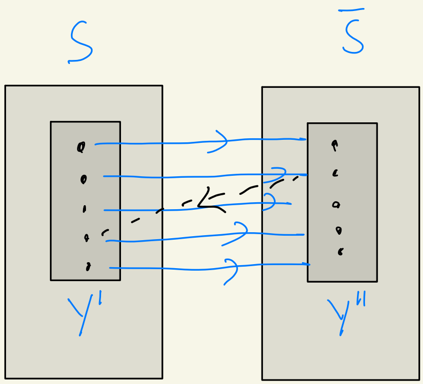
$\Updownarrow$

booking $i$ precedes booking $j$ by at least 5 minutes

- $D$ is clearly acyclic as every path $v_{i_1} \to v_{i_2} \to \dots \to v_{i_r}$ in $D$ corresponds to bookings $b_{i_1}, b_{i_2} \dots b_{i_r}$ that may be handled in one tent in that order
- Hence a path cover $P_1, P_2 \dots P_k$ of $D$ corresponds 1-1 to an assignment of the bookings in $S$ to tents
- We can handle all bookings $\Leftrightarrow$ $D$ has a path cover with at most $k$ paths where $k \leq K$ (# of tents)
- We have seen in 4.1 that this can be modelled as a minimum value flow problem in a network $N_D$ (as in 4.1) so there is a solution to the bookings problem if and only if the minimum value of an $(s,t)$-flow in $N_D$ is at most $K$
- If this value is larger than $K$ then we can find a set of at least $K+1$ pairwise overlapping bookings by considering an $(s,t)$ cut $(S, \bar{S})$ with

$$\ell(S, \bar{S}) - u(\bar{S}, S) > K$$




$(S, \bar{S})$ cut with max demand

- no arc inside $Y$ in $D$
- all bookings $\Leftrightarrow$ $Y$ 'overlap' pairwise

  no two can be in the same tent

- $|Y| = \ell(S,\bar{S})$
  $\geq \ell(S,\bar{S}) - u(\bar{S},S)$
  $=$ min value $(s,t)$-flow
  $=$ min # tents required
  $> K$

b) We showed above that

$K' = \min$ # tests needed $= \min\{|x| \mid x$ feasible in $N_D\}$

So we can find $K'$ by solving the minimum value $(s,t)$-flow problem in $N_D$ and this can be done by 2 max flow calculations by BJG sect 3.9.

c) Let each the arc $(v' \to v'')$ in $N_D$ have a cost of $-1$ and change its lower bound to 0. Then $y \equiv 0$ is feasible in new $N_D'$ and every flow $x$ of value $r \leq K$ and cost $C$ corresponds to a legal assignment of $C$ bookings to $r$ tests.

Hence we solve the problem by finding a minimum cost flow of value $K$

As $x \equiv 0$ is feasible and since $N_D'$ is acyclic and hence has no negative cycle, we can use the buildup algorithm

- Each iteration will increase the flow value by 1 so the total # of iterations is $K$ and each takes $O(nm)$

Hence the complexity is $O(Knm)$

Here $n \in O(p)$ as $N_D$ has 2 vertices for each booking and

$m$ is the number of arcs in $N_D$ which is proportional to the number of pairs of non overlapping bookings

so we can write the complexity as $O(Kpm)$

- If we run cycle cancelling we first find a flow $x_0$ of value $K$ in time $O(Km)$ by Ford-Fulkerson and then augment along negative cycles as long as one exist. The number of iterations is at most $p = |S|$ as $cx_0 \leq 0$ and every $(s,t)$-flow has cost at least $-p$ so complexity is $O(pnm) = O(p^2m)$

Buildup is $O(Kpm)$ so as $K << p$ we see that buildup is the best.

d) Let $x_k$ be a min cost flow from c)

and find $x_{k+1}, x_{k+2}, \dots x_{k'}$ via

the buildup algorithm.

Then each $x_{k+i}$ is optimal by the

buildup property so $-c(x_{k+i})$ is

exactly the maximum # of

bookings we can handle by

opening $i$ new tents.

The complexity is the same as running

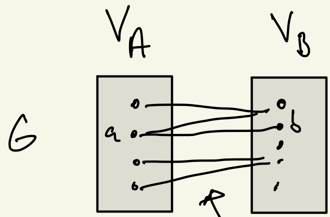
the buildup alg for $k'$ iterations

so it follows from the answer to c)

that the complexity is

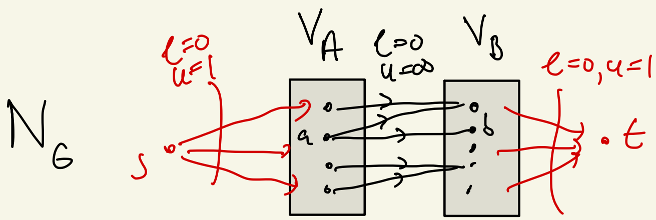
$$O(k'pm)$$

# Problem 6




edge in G is ab is a good pair

- G is bipartite and a good collection $a_1b_1, \dots, a_pb_p$ is exactly a matching in G
- Hence finding a maximum sized good collection is equivalent to finding a maximum matching in G



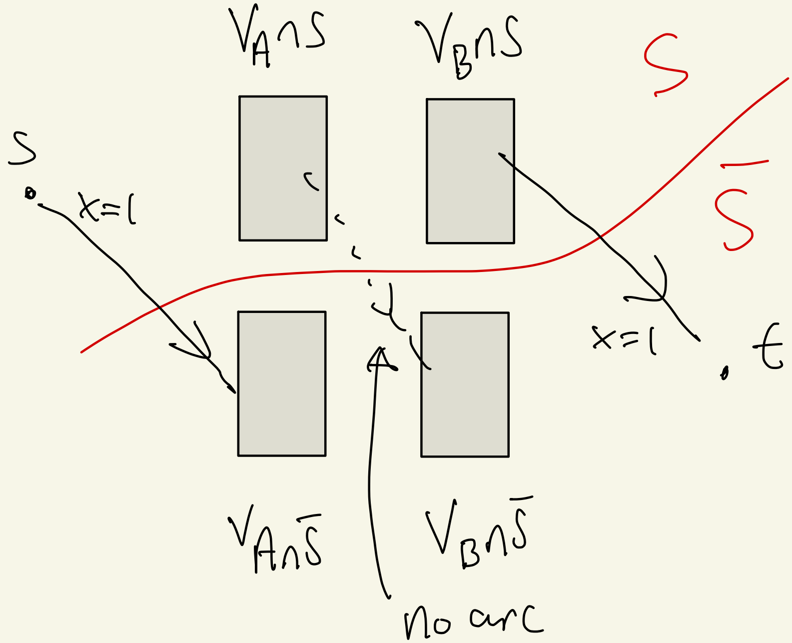

- We have seen several times that

$$\max\{|M^*| \mid M^* \text{ matching in } G\} = \max\{|x| \mid x \text{ is an } (s,t)\text{-flow in } N_G\}$$

So we can solve the problem by finding a maximum (integer valued) flow $x$ in $N_G$ and returning the edges $ab$ for which $x_{ab}=1$

- We have seen that we can certify a maximum matching of size $k$ by a vertex cover of size $k$

look at the cut $(S,\bar{S})$ when $x$ is a max flow and $S = \{v \mid \exists (s,v)\text{-path in } N(x)\}$

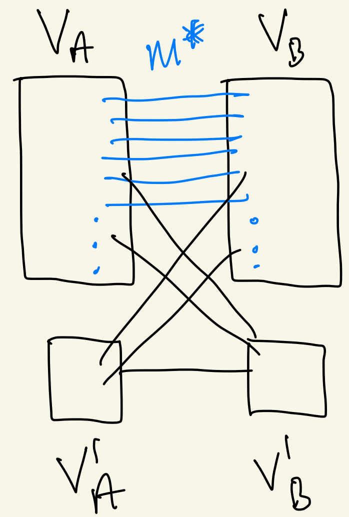



$$
\begin{aligned}
|M^*| &= |x| \\
&= u(S,\bar{S}) \\
&= |V_A \cap \bar{S}| + |V_B \cap S| \\
&= |W| \quad \text{when } W = (V_A \cap \bar{S}) \cup (V_B \cap S)
\end{aligned}
$$

and $W$ is a vertex cover

b) $p = |M^*|$ assume $p$ too small

$G'$




$k = |V'_A| = |V'_B|$

Consider $N_{G'}$ build as $N_G$ and our maxflow $x$ in $N_G$. Then $x$ is feasible in $N_{G'}$ and the maxflow value in $N_{G'}$ is at most $|x| + 2k$. So we can find a maxflow in $N_{G'}$ by using at most $2k$ augmenting paths in $N_{G'}(x)$. These can be found in time

$$O(k(|V'| + |A'|)) \qquad V' \text{ vertex set of } N_{G'}, \; A' \text{ arc set} \ldots$$

$$= O(k(n' + m'))$$

c) Let $M^*$ be the

good collection of size $p$ that we found in a) and let $p' > p$ be the size of a max matching in $G'$

Assign costs to $A'$ as follows:

- the cost is $-1$ if the arc is of the form $a \to b$ when $ab \in M^*$
- all other arcs get cost 0

Then the matching $M'$ corresponding to a min cost flow $x'$ of value $p'$ shares as many edges with $M^*$ as possible