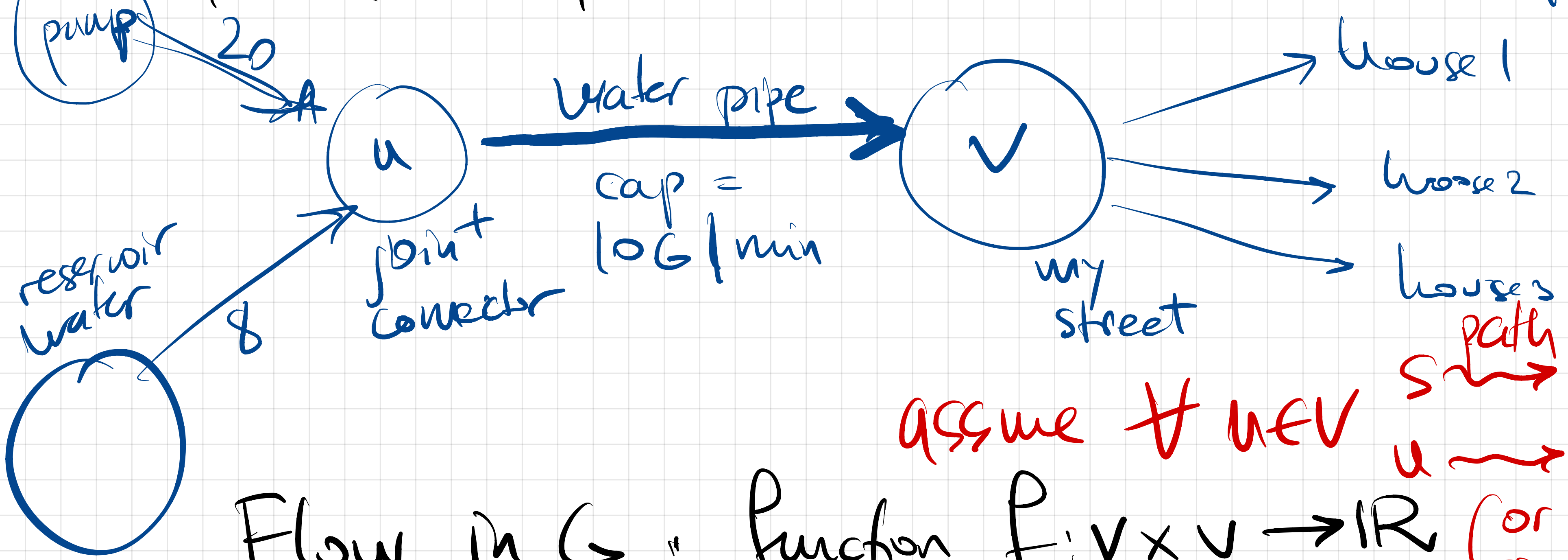
$\boxed{\text{Lecture 13}}$ Network Flows (Graphs Part 3)

- next week: last lecture (hope in class) no examination + attendance
- HW 11 due now
- HW 12 due next Wed. (and Push Relabel Demos)
- FINAL exam Sat 12/11 10AM–4PM Skillman 3rd fl
  4 TAs proctors + me online
- TRACE evals: evaluate course, instructor, TAs, faculty, examination etc. Please be honest both good / bad

Graph + Edge Capacities (Directed) ⇒ "Directed water pipes"

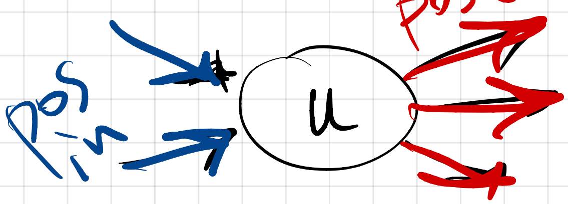


assume $\forall u \in V$ path $s \rightsquigarrow u$, $u \rightsquigarrow t$ (or else remove u)

Flow in G: function $f : V \times V \to \mathbb{R}$

$f(u,v)$ = how much water flows from u to v

- $f(u,v) \le w(u,v)$ at most edge capacity
- symmetry $f(u,v) = -f(v,u)$ no big deal, just for math
- conservation at each vertex



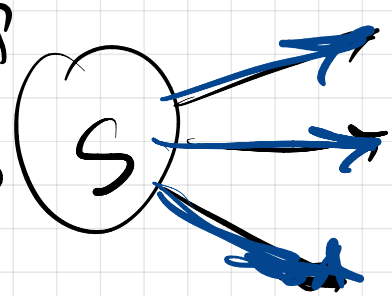
$$\sum_{\substack{v \to u \\ f(v,u) > 0}} f(v,u) = \sum_{\substack{u \to z \\ f(u,z) \text{ pos}}} f(u,z)$$

post neg target

$$\sum_{z} f(u,z) = 0$$

- exception: 2 special vertices s = source t = sink

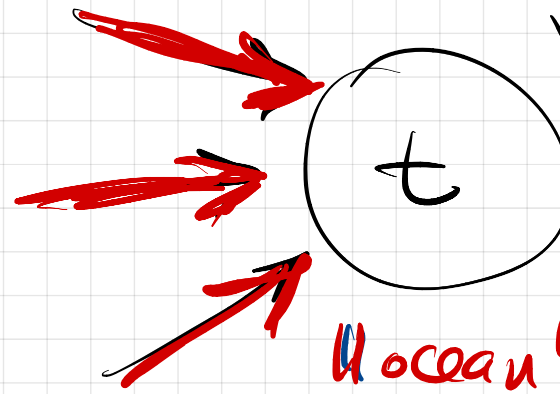
nothing coming in



"river source on mountain"

$\boxed{\text{only positive-out flow}}$

conservation does not apply to s, t

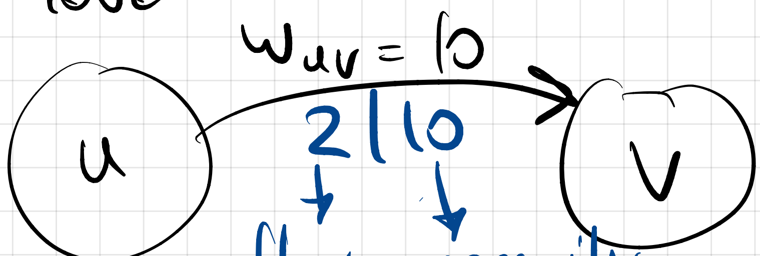


nothing going out

"ocean"

$\boxed{\text{only positive-in flow}}$

- in G look at a directed edge $u \to v$

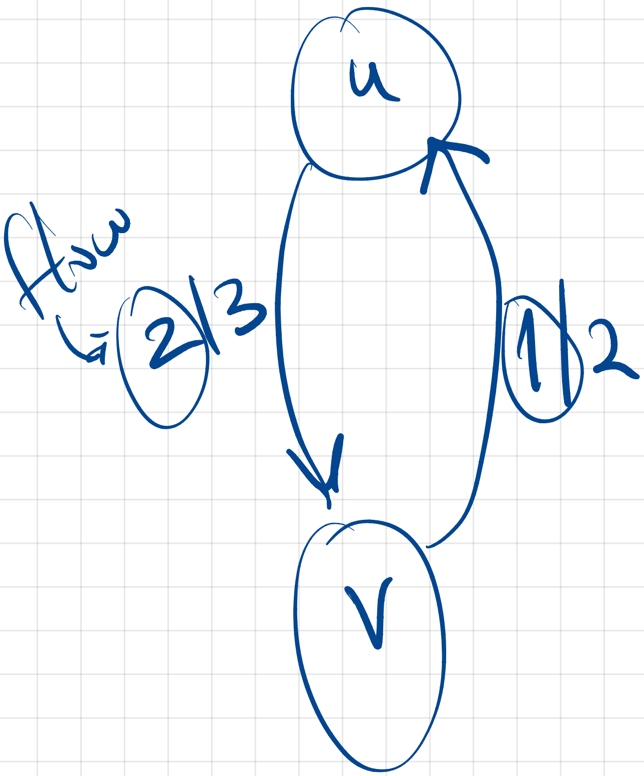


physical vs model
(actual water) (theory)

physical / model

- can flow at most 10 units of flow from u to v
- cannot flow water from v to u.

model/theory

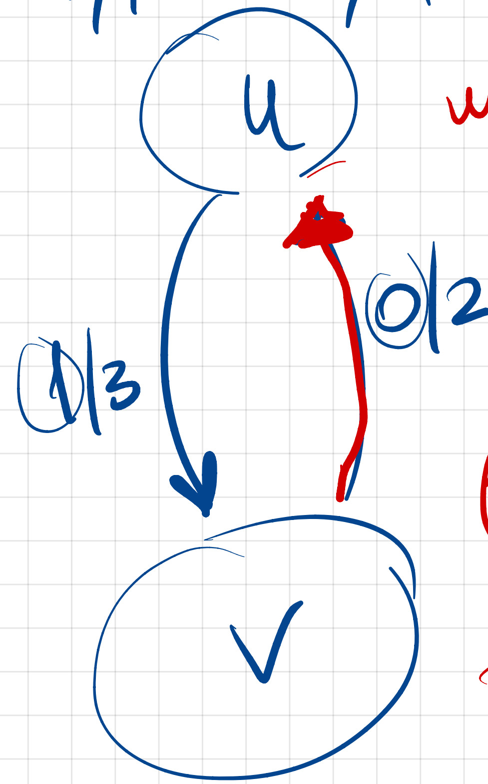


flow cancellation (theory)

model still valid due to cancellation

$f(u,v)=2$
$f(v,u)=1$

reality/practice/physical



might not exist in ↳ (not nec for model)

$f(u,v)=1$
$f(v,u)=0$

$\boxed{\text{Task}}$: send max flow from s to t

flow capacity $|f| = \sum_u f(s,u)$ total out from source $= \sum_u f(u,t)$ total in to the sink

Network Flow Alg: Ford-Fulkerson

want max flow

- initialize flow $f=0$ on all edges

loop

- find "augmenting" path $s \rightsquigarrow t$ that admits flow

$$s \to u_1 \to u_2 \to \cdots \to u_k \xrightarrow{V} u_{k+1} \to \cdots \to t$$

admit flow: $\forall$ edge rem. capacity $w(u_k u_{k+1}) > 0$ — after: $V - V = 0$

bottleneck

$V = \min$ edge cap on path. $V > 0$

- send V units flow on that path — flow increase $+V$
  - update flow $f_{edge} = f_{edge} + V$
- update path-edges capacities $w = w - V$ → also create new edges! (did not exist in G)
  (min edge becomes 0) "saturation"

Reset

Q1: how does it work / implement?

Q2: proof

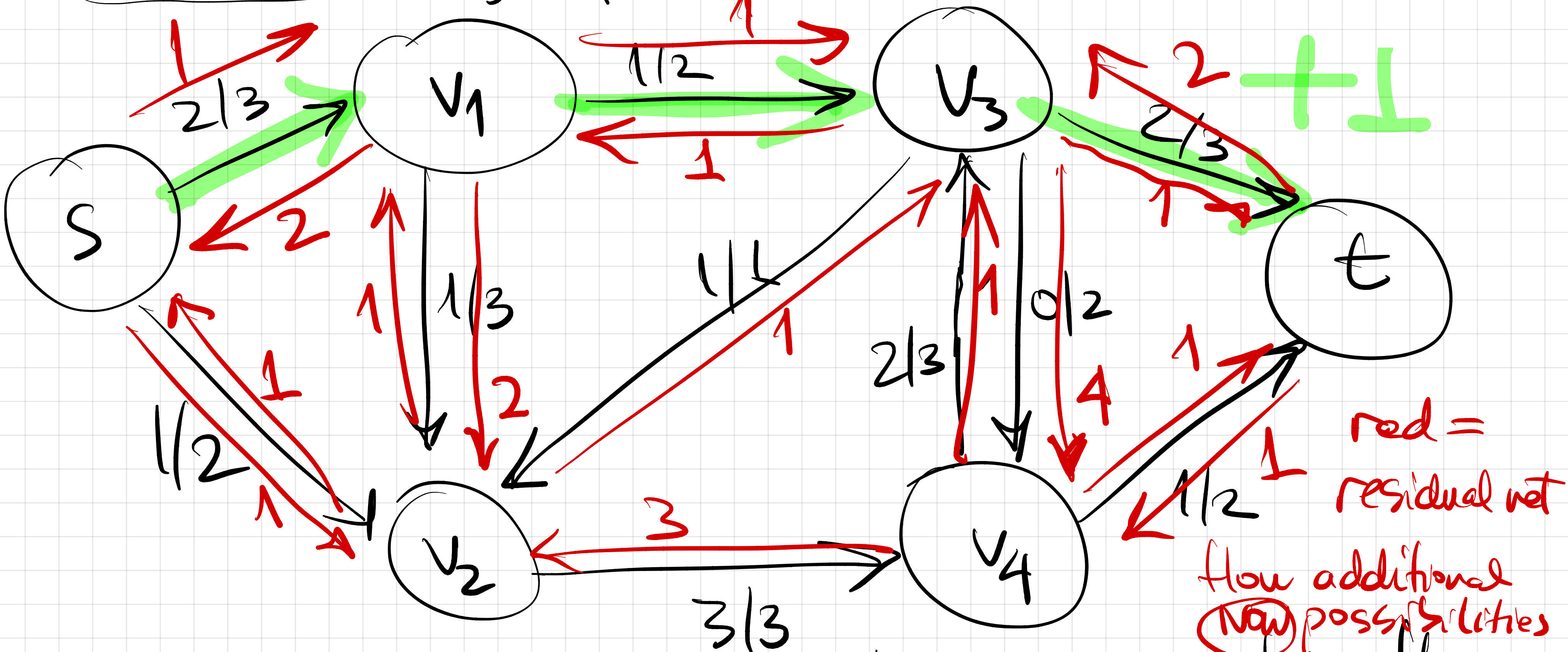
Example

$\xrightarrow{2|10}$

2 = current flow
10 = edge cap.

(in the middle) situation

Current |flow| = 3

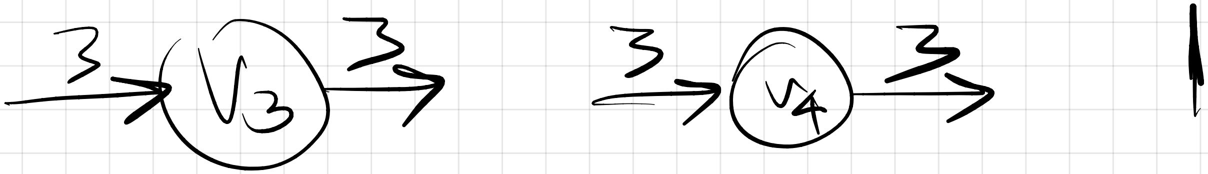


- current flow valid? ✓
  - f(u,v) ≤ edge cap ✓
  - conservation for each vertex (not s,t)

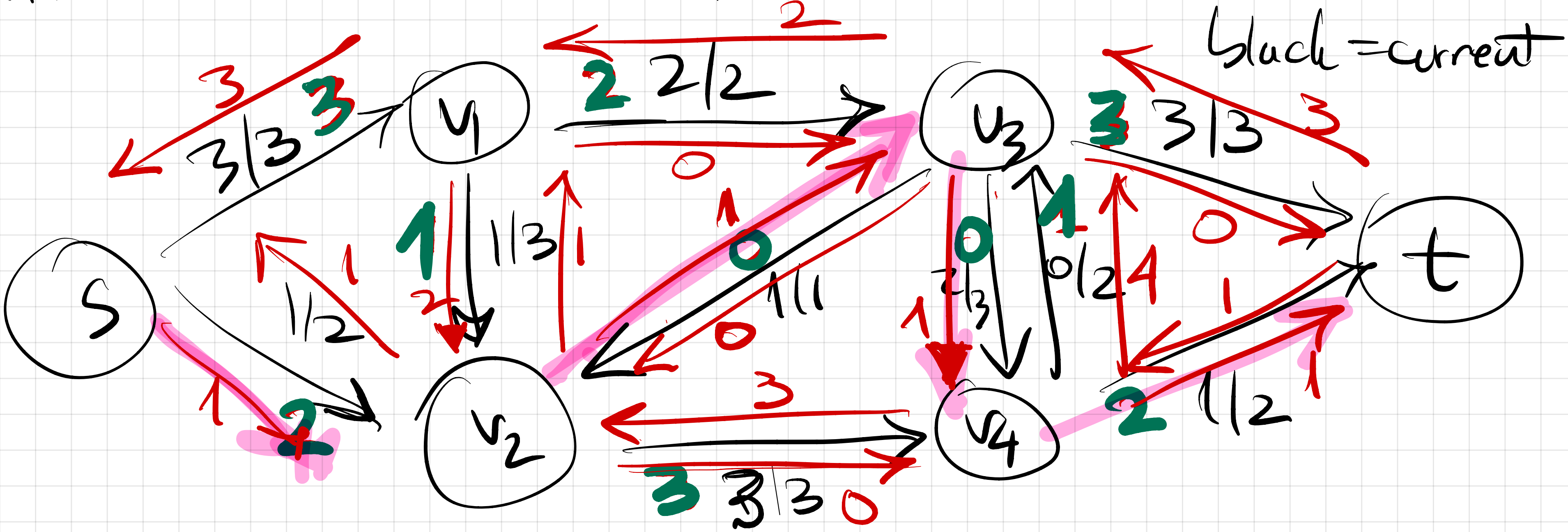
$\xrightarrow{3} V_2 \xrightarrow{3}$ $\qquad$ $\xrightarrow{2} V_1 \xrightarrow{2}$

- can I send more flow (find augmenting path)

$s \xrightarrow{1} v_1 \xrightarrow{1} v_3 \xrightarrow{1} t$



after that additional flow $s \to 1 \to 3 \to t$ +1



total flow = "from source" 4

- can I find another path to send more flow?

Black → No | flow = 4

no other path ⇒ 4 final flow?

Red: Yes $s \xrightarrow{1} v_2 \xrightarrow{1} v_3 \xrightarrow{1} v_4 \xrightarrow{1} t$

- max flow in this graph (from scratch)

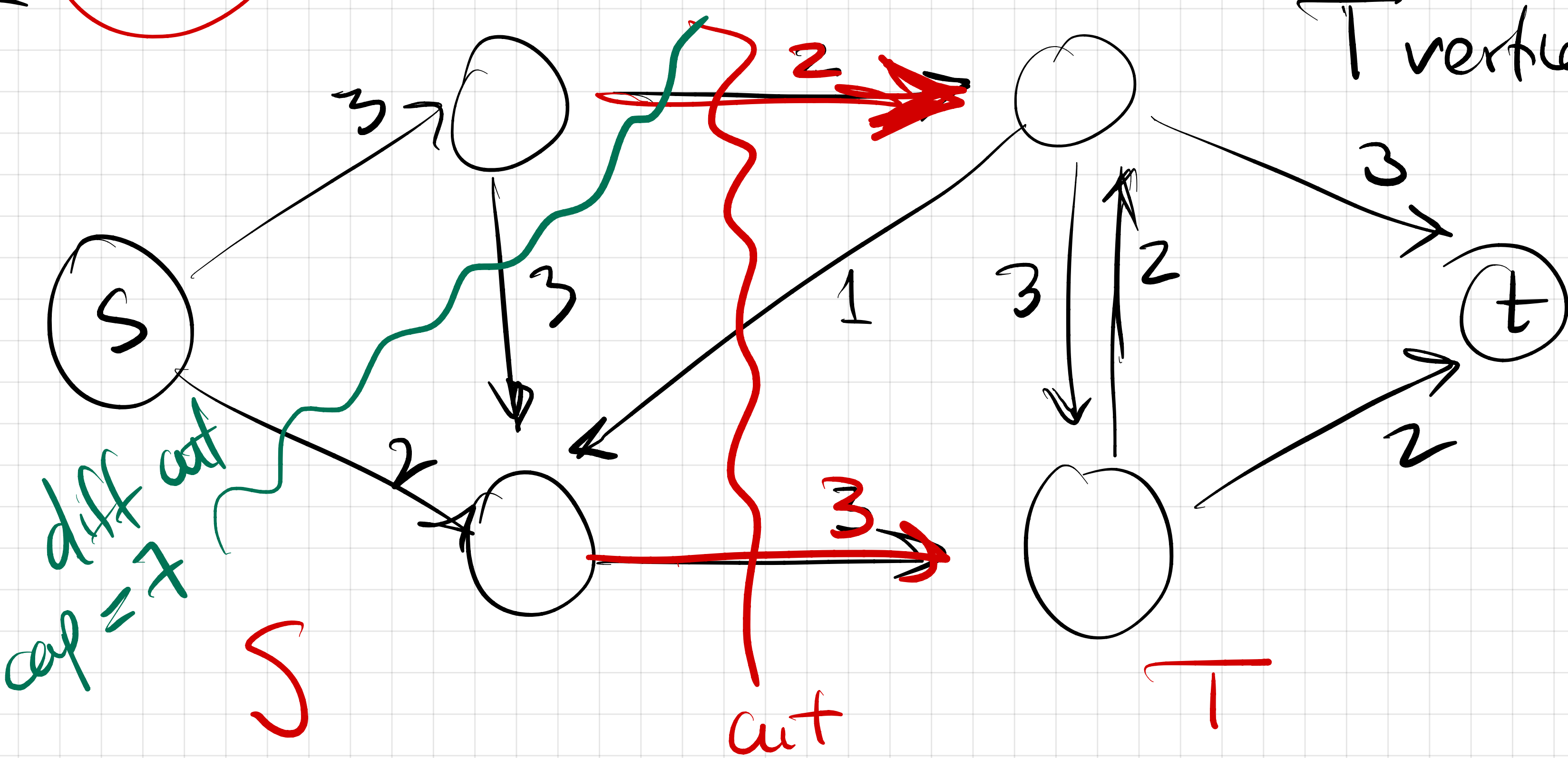
only green flow is 5

$v_2 \to v_3$ not edge in G

stop the $v_3 \xrightarrow{1} v_2$ water

Q2. Proof that alg FF works.

def Cut in the network partition S vertices, inc s
T vertices, inc t



capacity of the cut $S,T = \sum_{\substack{u \in S \\ v \in T}} w_{uv} = 2+3 = 5$

$\forall$ cut cap $\geq$ $\forall$ flow size

Max Flow – Min Cut theorem

OBS: f valid flow, S,T cut cap C then $|f| \le C$

(a) (b) (c) equivalent statements

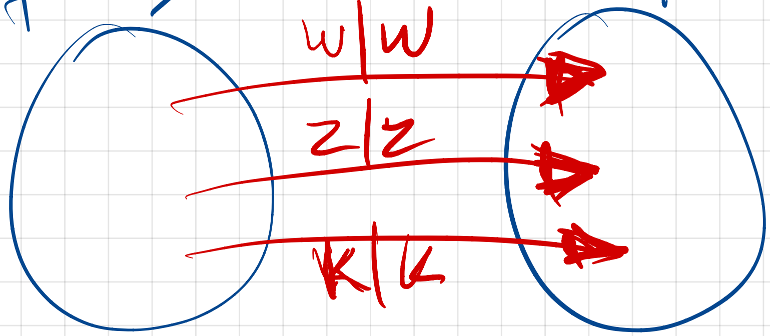
(a) f max flow

(b) residual net R_f has no augmenting path
incl. new red edges.

(c) there is a cut S,T saturated by the flow $|f| = \text{cap}(S,T)$

proof: (a) $\Rightarrow$ (b) already done:
if there's add aug path $\Rightarrow$ increase the flow (not possib if f maximal)

(c) $\Rightarrow$ (a) if S cut T saturated $\Rightarrow$ f already uses all edges $S \to T$ at max cap



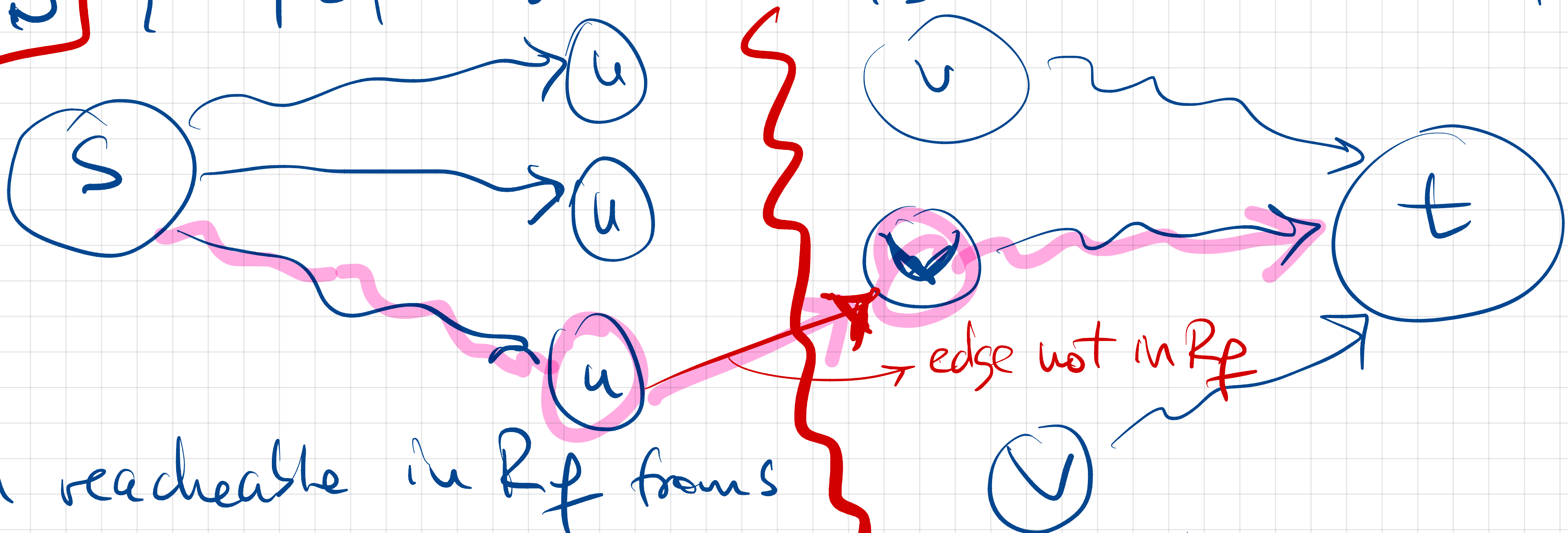
that flow cannot be increased

(b) $\boxed{\text{No aug path exists in } R_f}$ $\Longrightarrow$ (c) saturated cut S,T

Construct the cut S,T:

$\boxed{S} = \{ u \mid \exists \text{ path } s \rightsquigarrow u \text{ in } R_f \}$

$\boxed{T} = \{ v \mid v \rightsquigarrow t \text{ path exists in } R_f \}$



u reachable in R_f from s

$\exists$ "augm path" from $s \to u$

z is not connected to either s,t

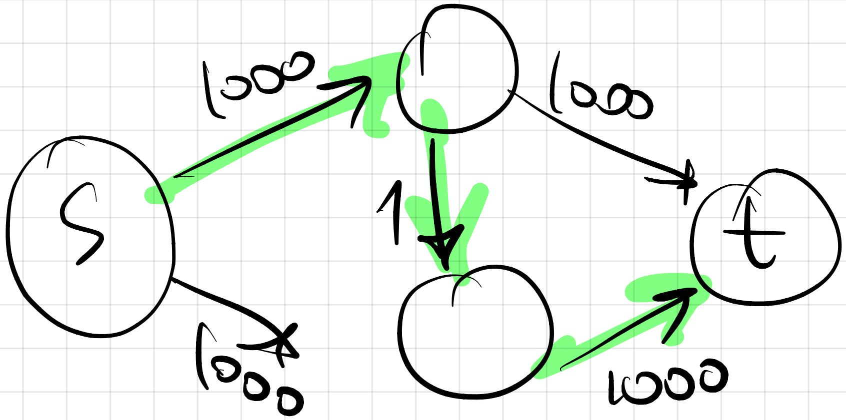
why cut S,T saturated?

in R_f (res. net) no edge avail $\boxed{S} \to \boxed{T}$

reason: if edge $u \to v$ exists in R_f $\Rightarrow \exists$ aug path $s \to u \to v \to t$

$\Longrightarrow$ cut saturated

Bad case for FF: choose a bad aug path repeat small size

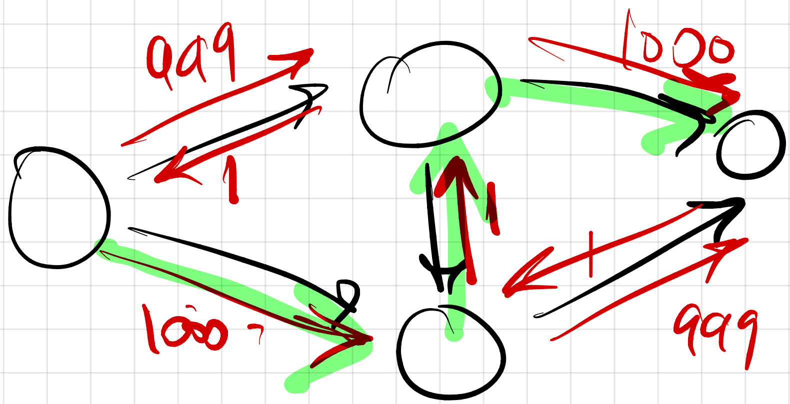


V=1

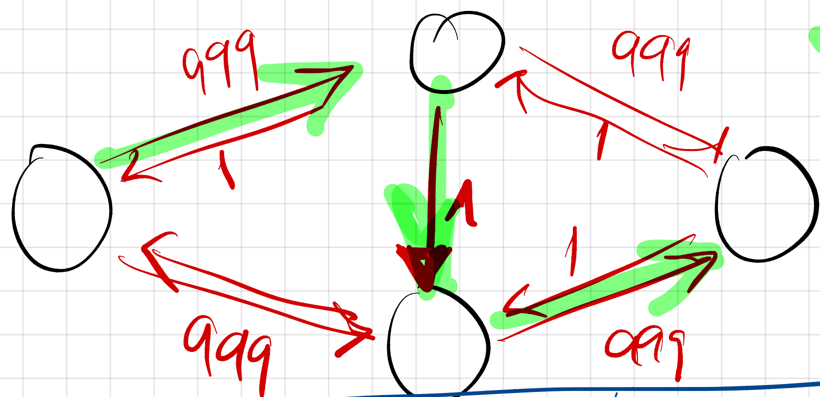
how many iterations if all aug path are bad (V=1)

2000!

max flow=2000



V=1



V=1

allow any aug path

DFS (aug path) worst case R.T.

$O(E \times |f|)$

$|f|$ = vague

best aug path / #itertations? (1) ; (2) 2 iterations

Solution (no proof): Edmonds-Karp.

- FF pipeline
- aug path by $\boxed{\text{BFS}}$ $\Rightarrow$ min # edges possible

$O(E)$

might not have the best capacity

$\Theta(V \cdot E)$ iterations (each edge can be min-critical V times)

BFS $\Theta(E)$ for the path

$\Theta(V \cdot E^2)$ worst case.

Much Better Net Flow Alg : Push Relabel (HW, not exam)

- same task: Find max flow (network s, t, edges w_{uv})

- flow very similar: $f(u,v) \le w_{uv}$ |flow| ≤ edge cap.

max flow = max out of the source.

symm: $f(u,v) = -f(v,u)$ math only

- conservation only at the end of alg

$$\sum_{\substack{v \to u \\ \forall v}} f(v,u) = \sum_{z \leftarrow u} f(u,z)$$

incoming to u

total outgoing from u

valid at the end.

- intermediary allow

$e(u) = \text{income flow}(u) - \text{outgoing flow}(u)$

$e(u) = \text{excess}(u) \geq 0$

u has a reservoir, can store some flow temporarily.

visual model

$V(x,y)$ fixed coordinates (x,y dont matter)

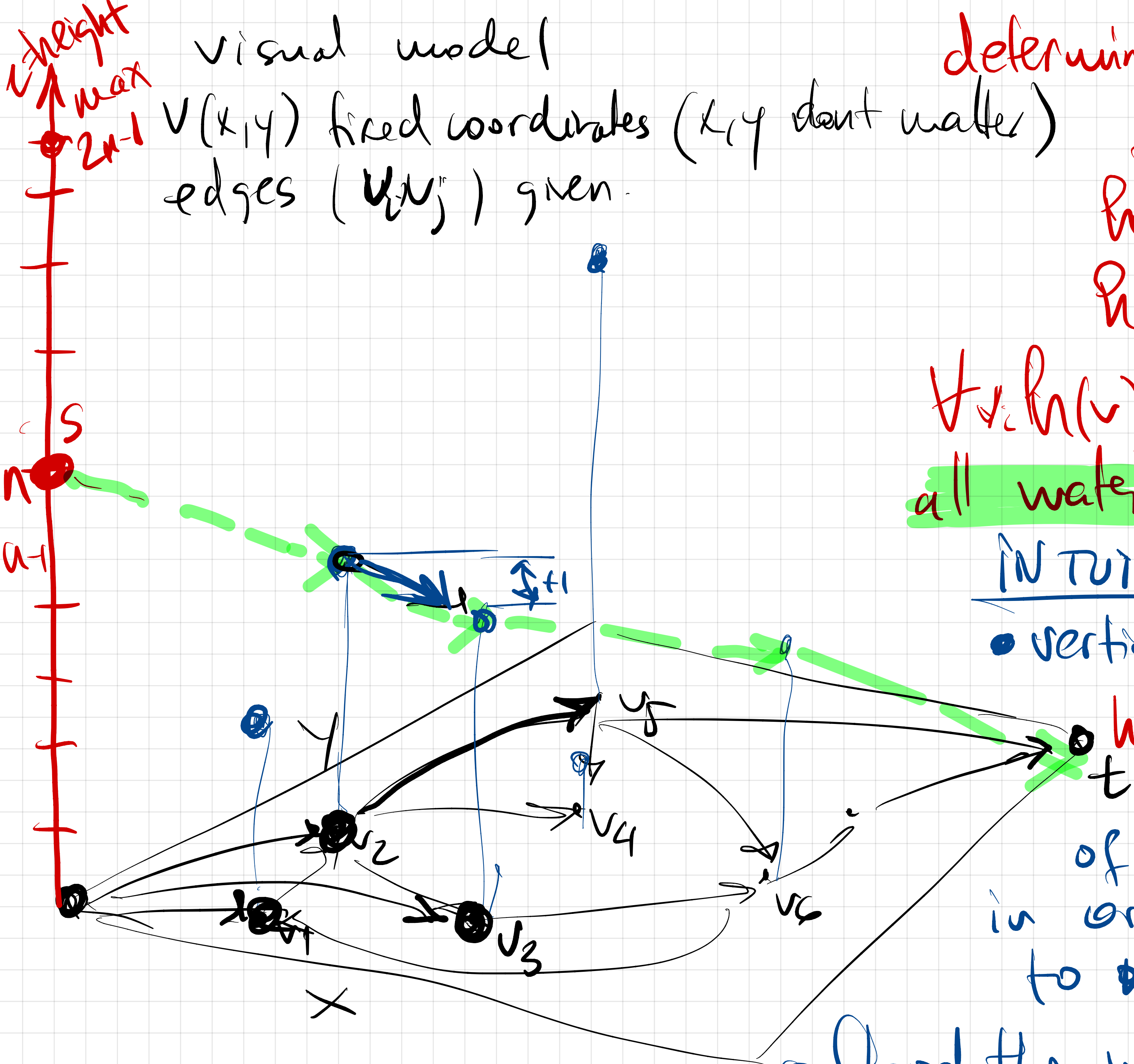
edges (v_i, v_j) given.

determine $h(v)$ height for every vertex

$h(s) = n$ fixed

$h(t) = 0$ fixed.

$\forall v: h(v) = ?$ s.t all water flows down



INTUITION:

- vertices closer to s with larger amount of water higher, in order to flow to next vertices
- flood the network in BFS-like stages.

Flood by stage → saturate all edges out of s

- Flood all vertices from the source 
  raise h(v) for those to flow water to others
- Flood 2nd wave with water from the first wave
  raise h(v) 2nd stage to the . . . — —
- Flood next vertices

details: raise(neigh) cycles possible

Push-Relabel alg.

init: $h(s)=n$; $h(v)=0 \ \forall v \neq s$

$e(s)=\infty$ $\quad e(v)=0 \ \forall v \neq s$

"active" vertex v

$e(v)>0$ has excess

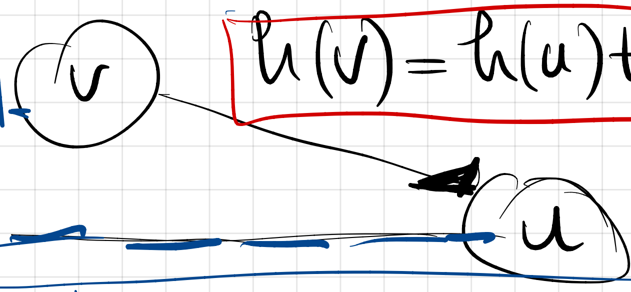
$e(v)$ need to be DISCHARGED $\Rightarrow$ water $e(v)$ must flow out of it.

$\exists$ edge $v \to u$ in Res. Net to discharge $e(v)$ excess flow.

Push flow from v

- $e(v)>0$ "active"
- edge to push $\boxed{res(v,u)>0}$
- heights $\boxed{h(v)=h(u)+1}$

$\Delta = 1$



Result: discharge $v \to u$ flow

$\min\{e(v), res(v,u)\}$

RELABEL "INCREASE HEIGHT"

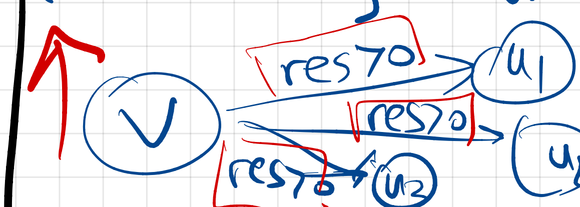
active excess flow $e(v)>0$

look for all avail res edges

$\boxed{res(v,u)>0}$

$\boxed{h(v) \text{ to low}}$

Result: more v higher, at the min-height where Push possible

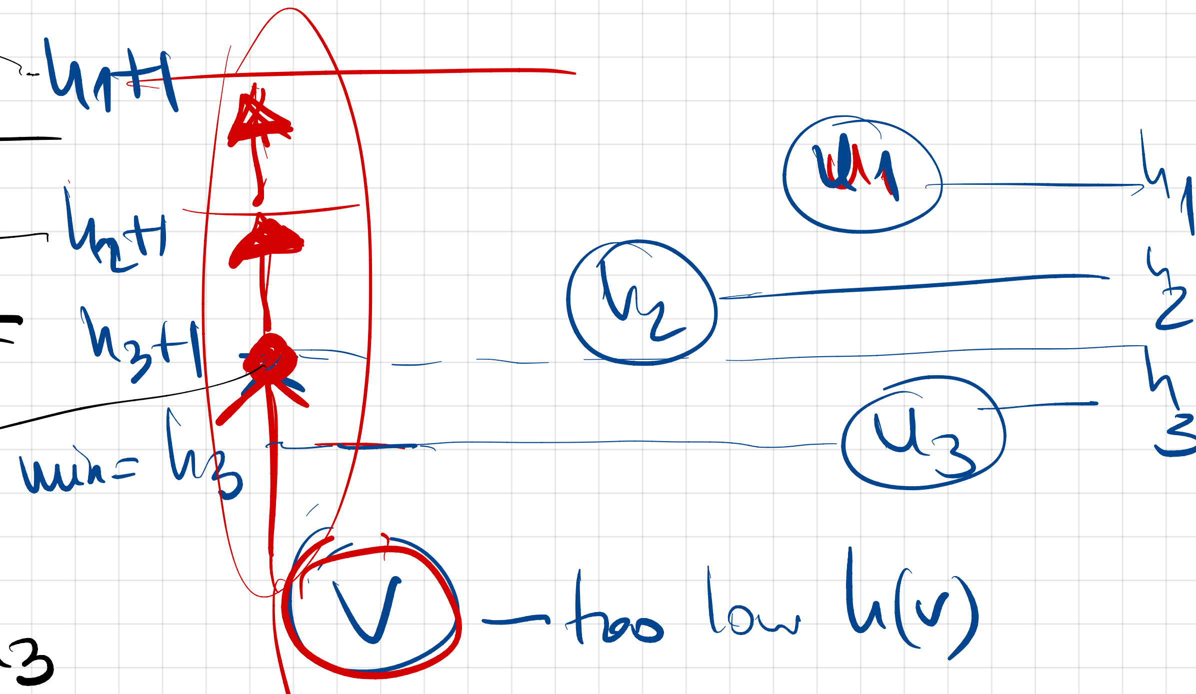


$h(v) = \min\{h(u)\}+1$

now V at level u_1+1
can push water
to u_1.

now v at $h=h_2+1$
can push down to u_2

V can
discharge/push
down to u_3
$h(v)-h(u_3)=1$



Relabel: height(v) goes up to min-level where it can push.

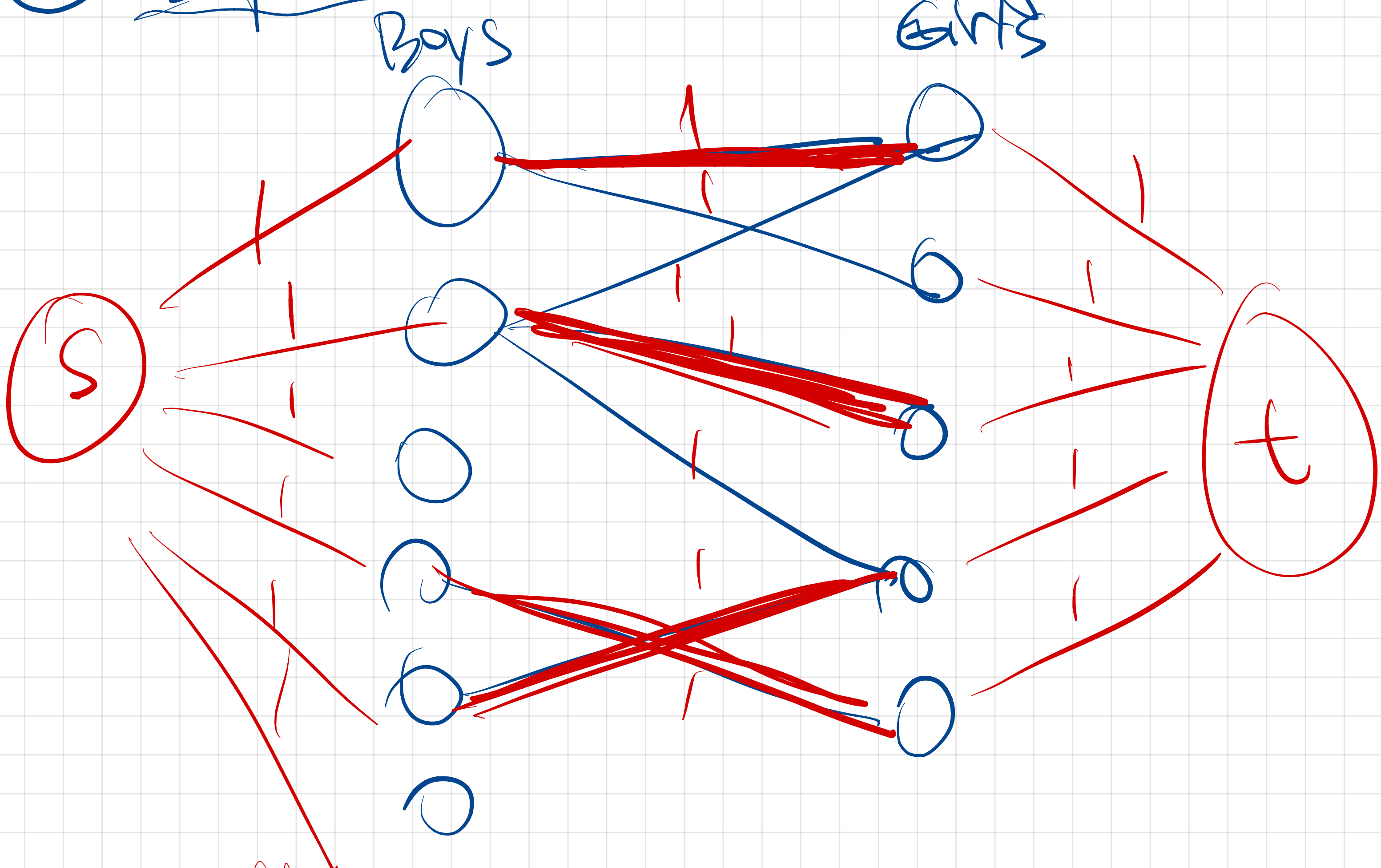
DISCHARGE FIFO

- pick active v $e(v)>0$
- discharge by repeated push/relabel until that v is not active ($e(v)=0$)
- only after consider other active nodes

$\Theta(n^3)$ very good

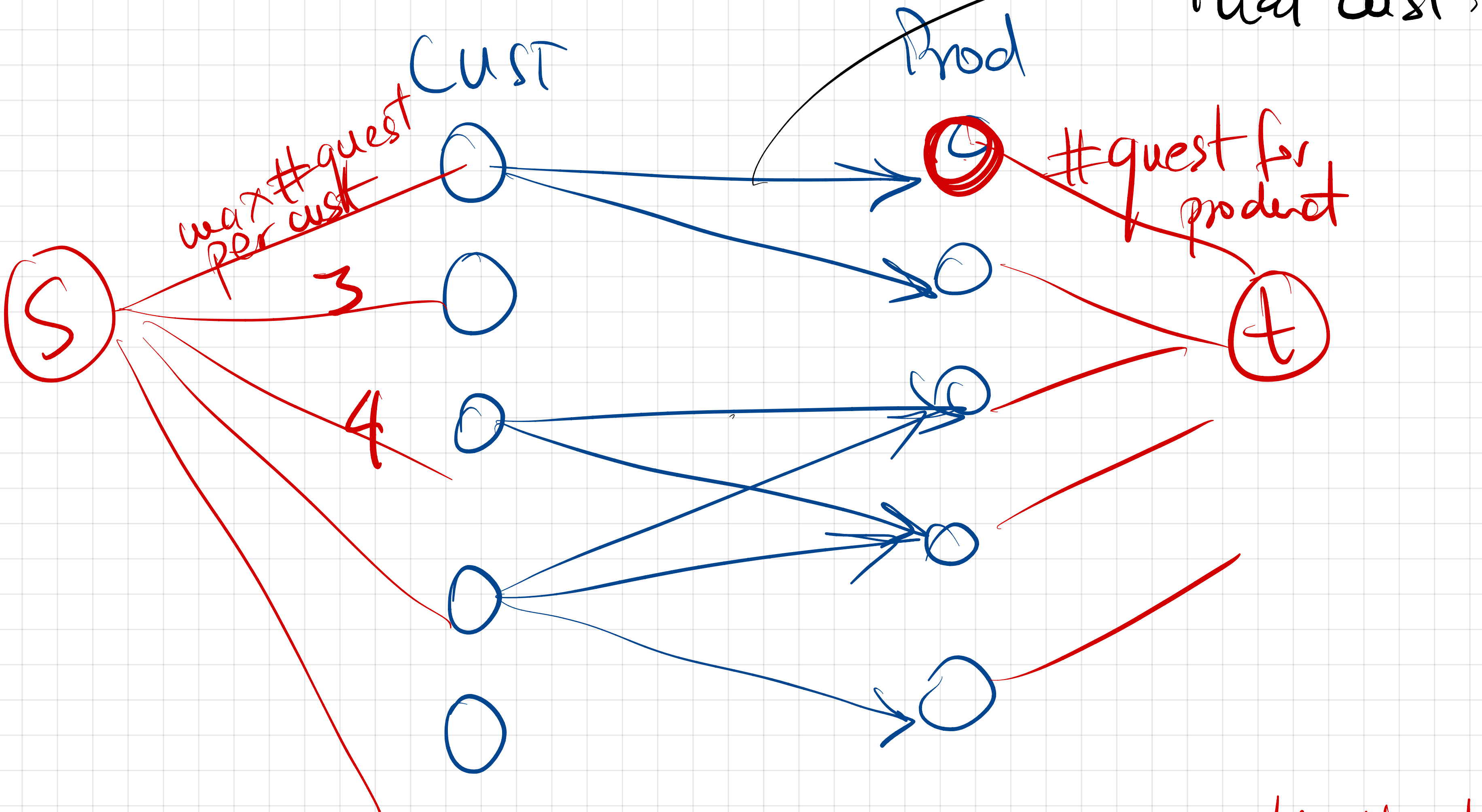
Unexpected Applications / Uses of Net Flows.

① Bipartite max matching   dancing max # pairs



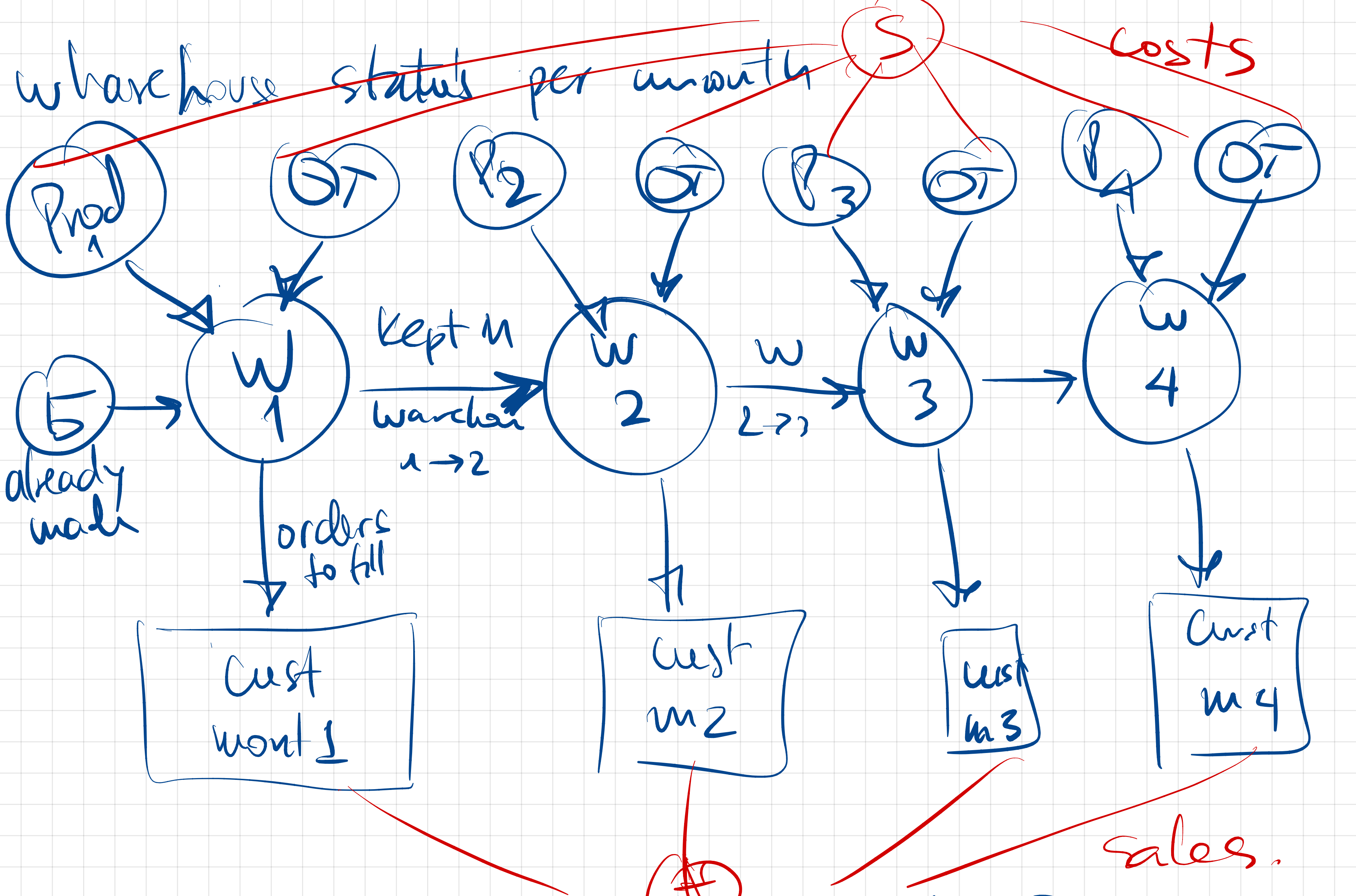
max flow ⟹ edges used are dancing pairs.

② Survey products x customers



max flow = best / max questions overall that can be asked.

③ wharehouse status per month



P_i = production of month i $\leq$ UB

OT_i = prod of month i not bounded) more expensive