21 Apr 2025 - Algorithms - Week 16

Single - source shortest path

Dijkstra $(w \geq 0)$

$(|E| + |V|) \log |V|$

Bellmann - ford

All pairs shortest path

① DP - 1 $\longrightarrow$ $|V|^3 \log |V|$

② DP - 2 $\longrightarrow$ $|V|^3$ $\longrightarrow$ Floyd - Warshall

$$g(i,j,k) = \min\{ g(i,j,k-1),\ g(i,k,k-1) + g(k,j,k-1) \}$$

③ Johnson's algo :

Idea : Run Dijkstra's algo $|V|$ times

only works for non-negative weights.

Idea 2 : modify weights to make them non-negative.
and preserve shortest path.

Step 1 : Run Bellman-ford from some fixed vertex s.

get negative edge cycles.

$\{ d(s,v) \text{ for } v \in V \}$

Time : $O(|V||E|)$

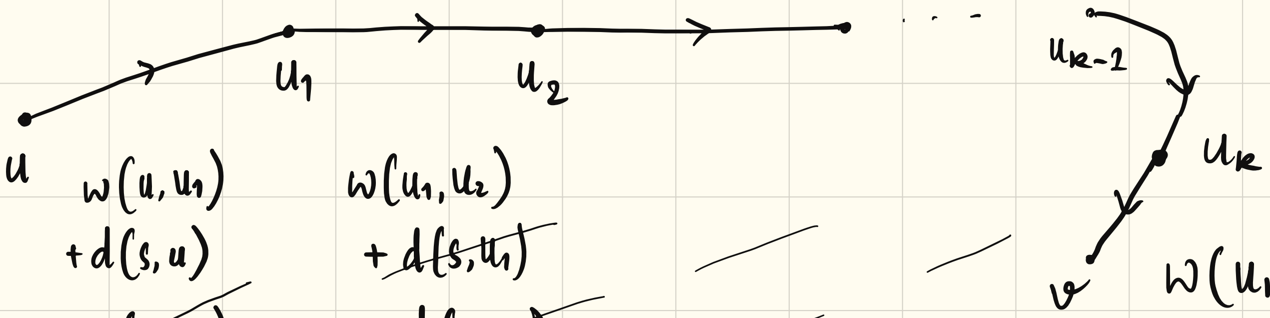
Step 2: $\hat{w}(u,v) = w(u,v) + d(s,u) - d(s,v)$

→ Why non-negative

→ How is shortest path preserved

$\hat{w}(u,v) \geq 0$

$\iff d(s,v) \leq w(u,v) + d(s,u)$ always true by Dijkstra's algorithm.



$$w(u, u_1) + d(s,u) - \cancel{d(s,u_1)}$$

$$w(u_1, u_2) + \cancel{d(s,u_1)} - \cancel{d(s,u_2)}$$

$$w(u_k, v) + \cancel{d(s,u_k)} - d(u_k, v)$$

$$\boxed{\hat{w}(\text{path}) = w(\text{path}) + d(s,u) - d(s,v)}$$

$$\hat{w}(p_1) - \hat{w}(p_2) = w(p_1) - w(p_2)$$

Step 3: Run Dijkstra from all sources using $\hat{w}$.

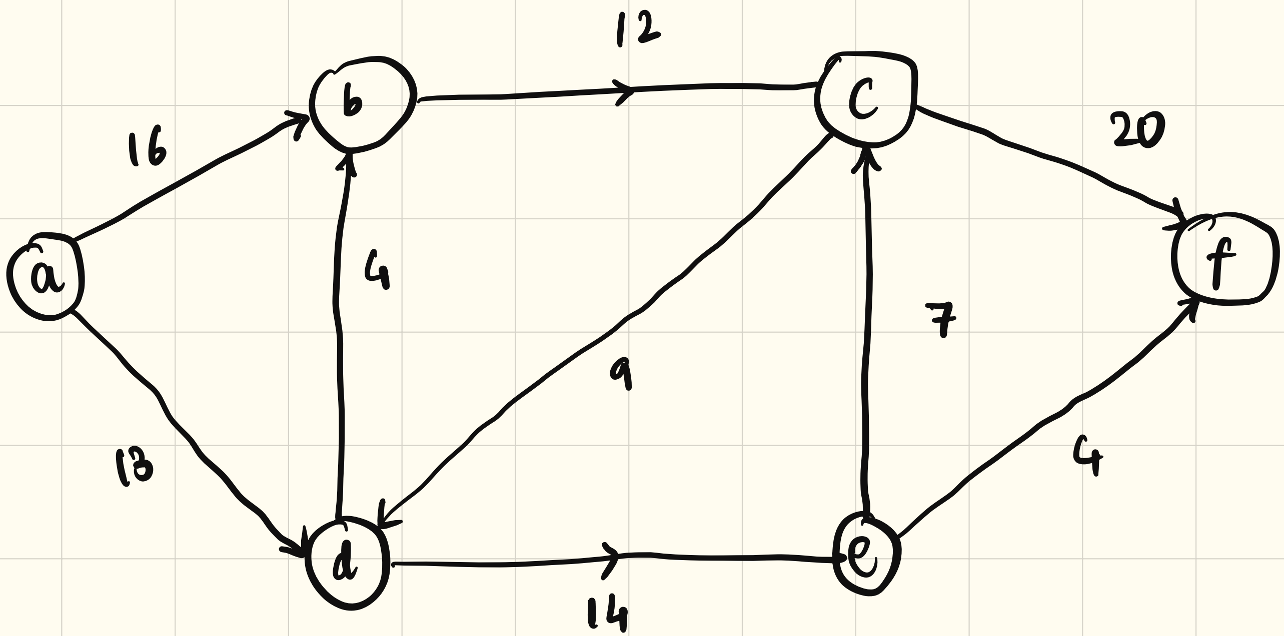
Time: $O(|V||E|)$ + $O(|E|)$ + $|V| \cdot (|E| + |V|) \log |V|$

Bellman-ford

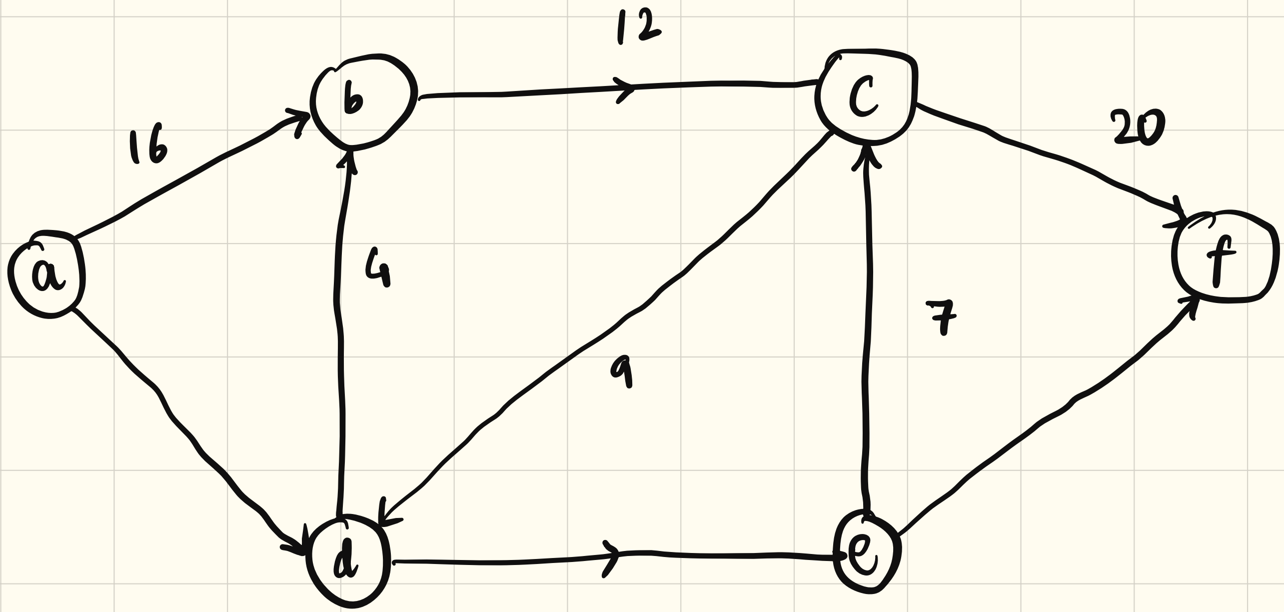
computing new weights

$$= O((|E| + |V|)\, |V| \log |V|)$$

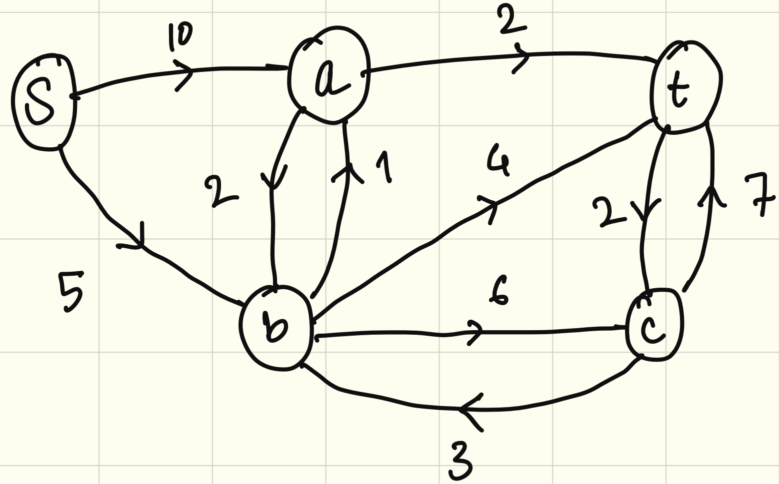
Max Flow Problem

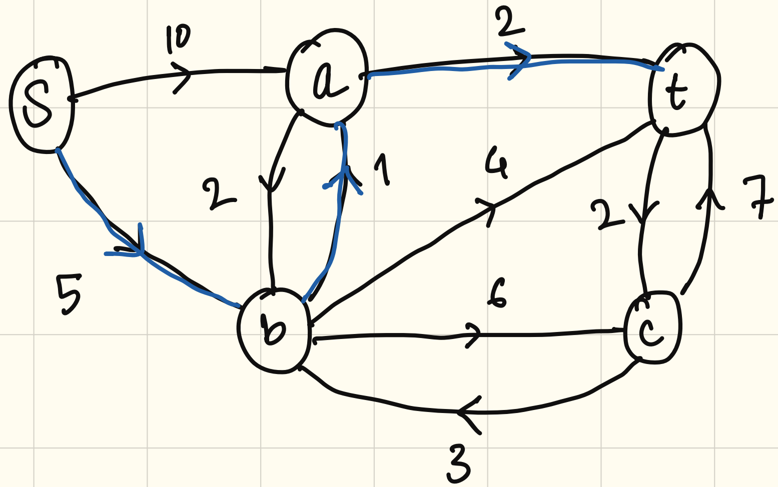


$a \to d \to e \to f$



Test
Test


S
a
t
b
c
10
2
2
1
4
2
7
5
6
3


S
a
t
b
c
10
2
2
1
4
2
7
5
6
3

Notes
Incomplete → Refer CLRS Chapter 24/26

4^{th} Ed prev.

24 Apr 2025

→ Correctness of Edmonds-Karp

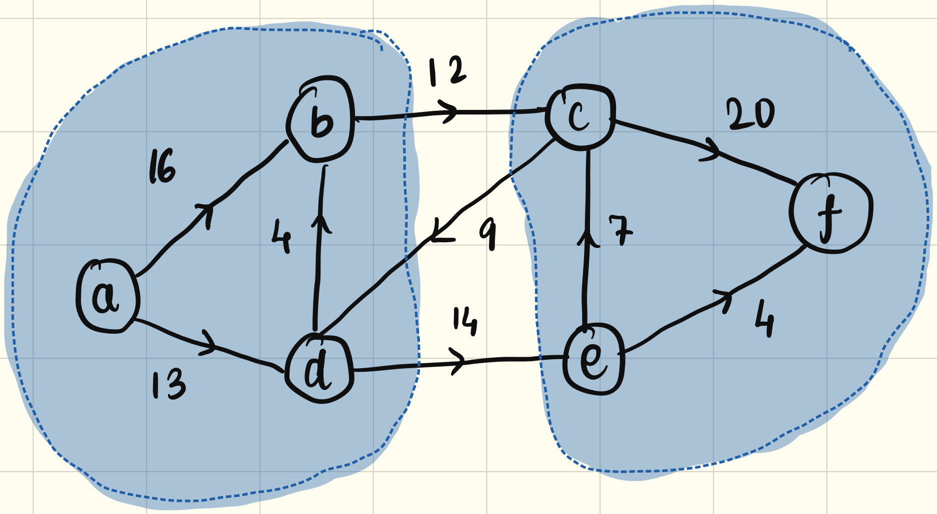
1. If G_f has no augmenting paths $s \leadsto t$,
   then f has a maximum flow value.

2.

① → Theorem (**Max-flow min-cut theorem**)

If f is a flow in G with source s, sink t, the following are equivalent:

1. $|f|$ is a max-flow in G
2. G_f has no $s \rightsquigarrow t$ path
3. $|f| = c(S, T)$ for some cut (S, T) of G.



$X = \{a, b, d\}$

$Y = \{c, e, f\}$

$c(X, Y) = 26$

$$c(X, Y) = \sum_{u \in X} \sum_{v \in Y} c(u, v)$$

$$f(X, Y) = \sum_{u \in X} \sum_{v \in Y} f(u, v) - \sum_{v \in Y} \sum_{u \in X} f(v, u)$$

s-t cut : (X, Y)

$s \in X, \; t \in Y, \; Y = V \setminus X$

Claim: $\forall$ s-t cuts X, Y

$$|f| = f(X,Y) \le c(X,Y)$$

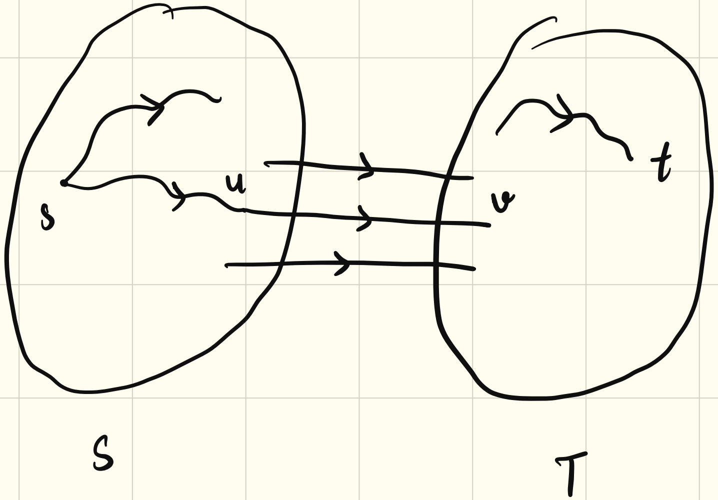
Intuitively, $\because$ flow is conserved

$$\textcircled{3} \Rightarrow \textcircled{1} \underset{\text{by contradiction}}{\Longrightarrow} \underbrace{\textcircled{2} \Rightarrow \textcircled{3}}_{\text{To prove}}$$

$S = \{ x : \exists\, s \rightsquigarrow x \text{ in } G_f \}$

$T = V \setminus S$

$t \in T$



Claim: $f(u,v) = c(u,v)$

If not: $f(u,v) < c(u,v)$

$\Rightarrow (u,v) \in G_f$

reachable from s

$\Rightarrow v$ is reachable from s $\Rightarrow\Leftarrow$

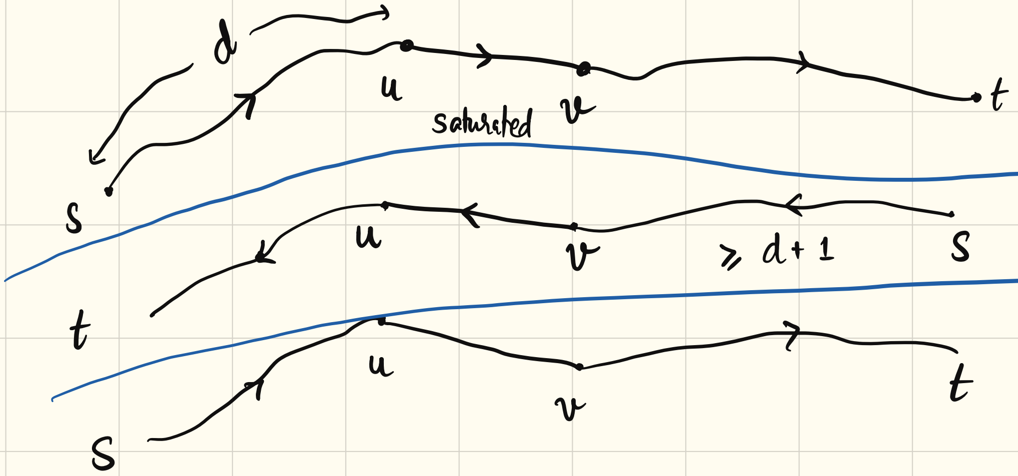
$\therefore \ |f| = c(S,T)$

② No. of steps taken by EK $\leq \frac{|V|}{2}$ (flow augmentation steps)

Claim: $\delta(s, v)$ is non-decreasing $\forall v$

An edge (u, v) is saturated by a flow if $f(u,v) = c(u,v)$

Claim: $\forall \ (u, v) \in E$, (u, v) gets saturated at most $\frac{|V|}{2}$ times.



$\delta(s, v) = d + 1$

$\delta(s, u) \geq d + 2$

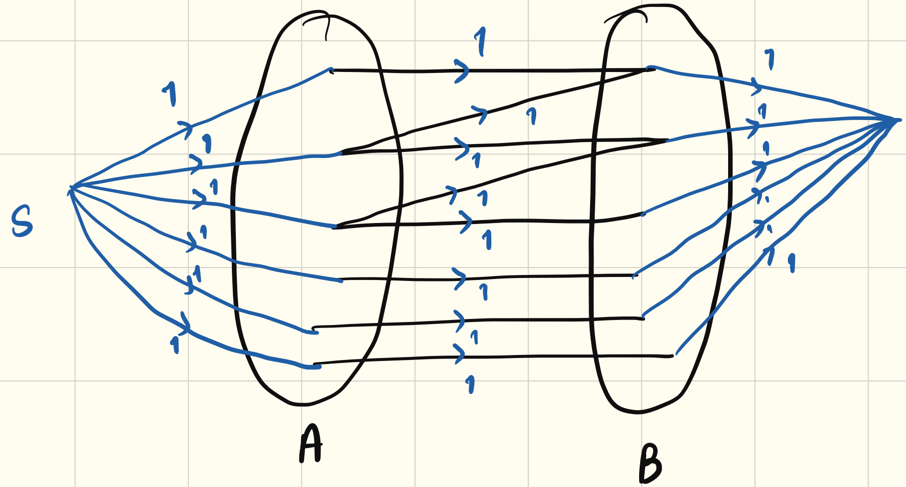
$\delta(s, u) \geq d + 2$

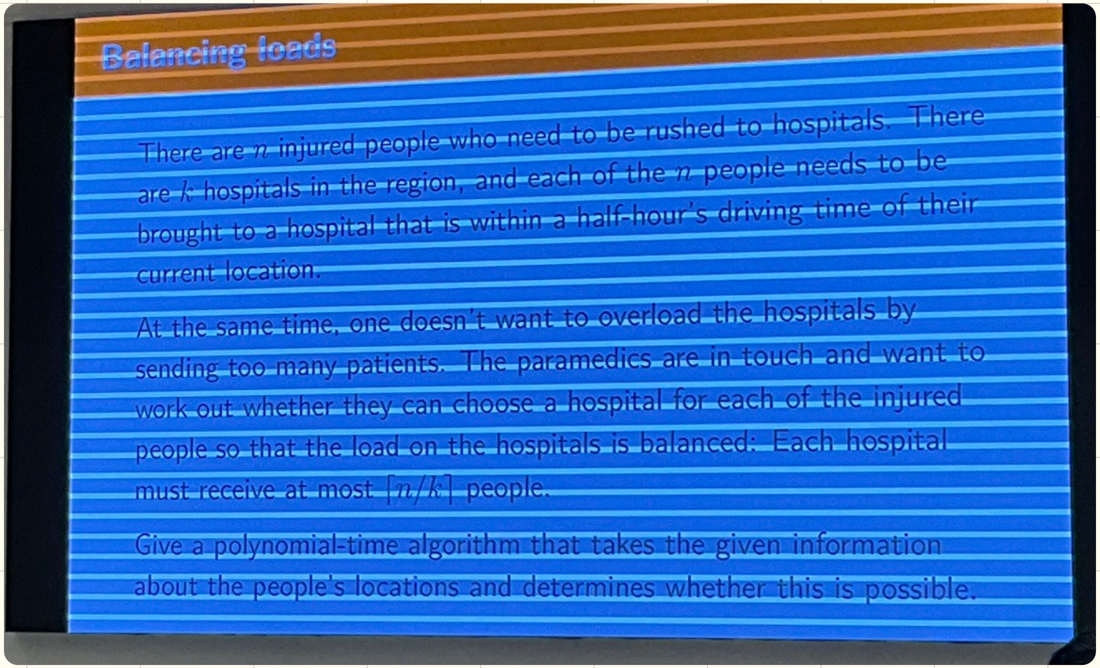
??

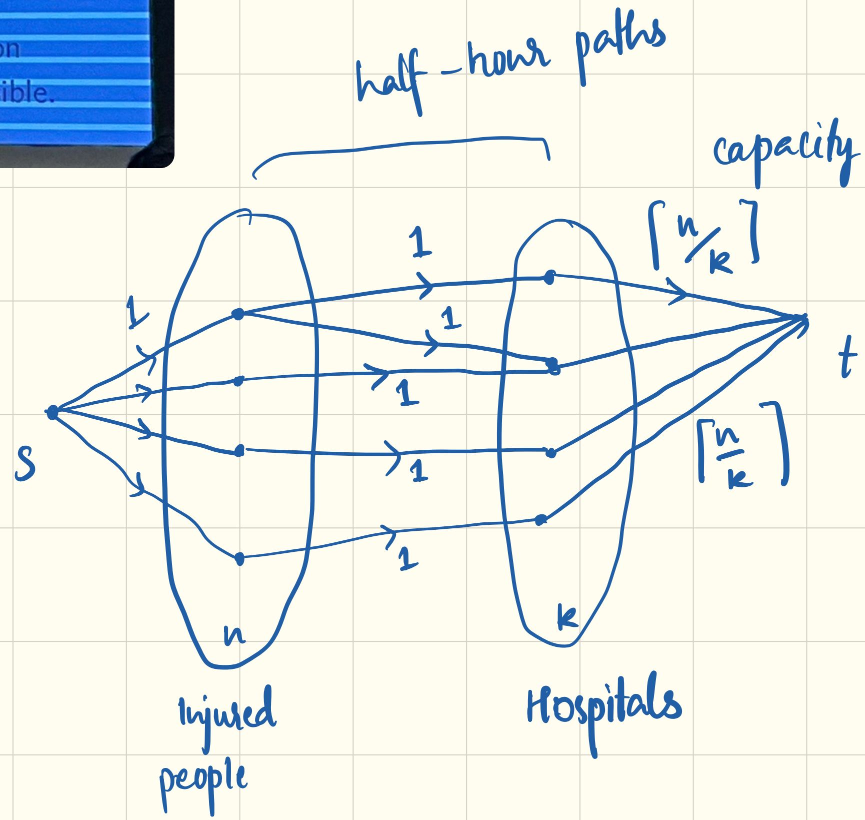
$\#$ of augmentations $\leq \dfrac{|E| \cdot |V|}{2}$

Running time of Edmonds Karp $= O(|E|^2 |V|)$

Maximum matching in bipartite graphs

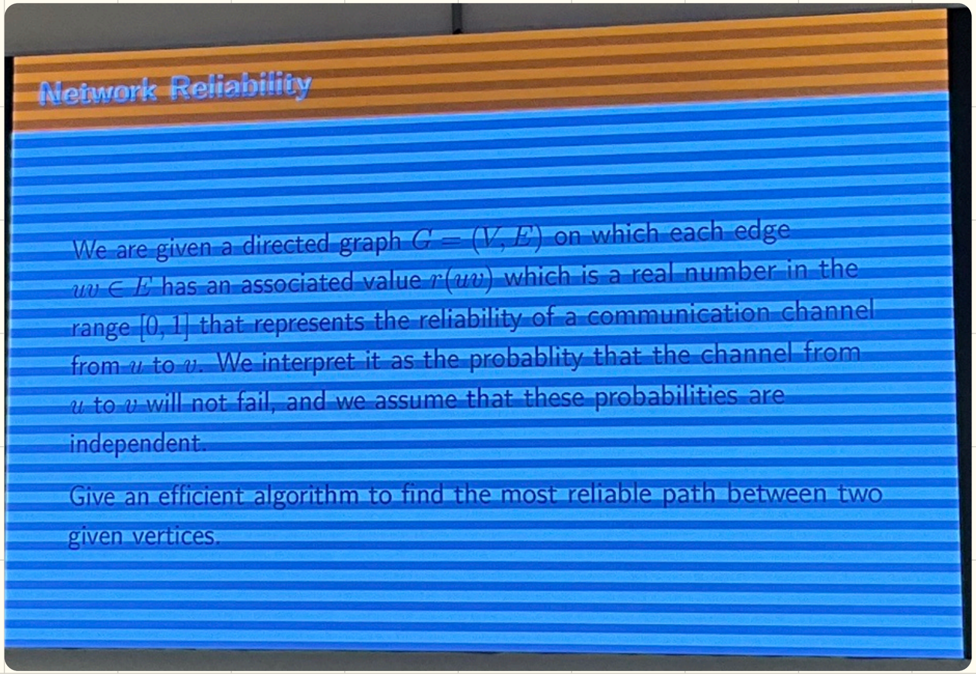



Balancing loads
There are n injured people who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location.
At the same time, one doesn't want to overload the hospitals by sending too many patients. The paramedics are in touch and want to work out whether they can choose a hospital for each of the injured people so that the load on the hospitals is balanced: Each hospital must receive at most $\lceil n/k \rceil$ people.
Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible.


half-hour paths
capacity
1
$\lceil n/k \rceil$
s
t
n
k
Injured people
Hospitals

Check if

$$|f| = n$$

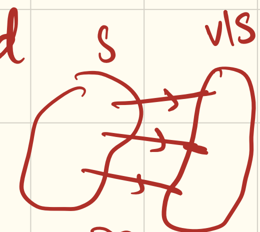


Brute-force: all paths from u to v

→ compute reliability (product)

→ return highest

Modified algorithm

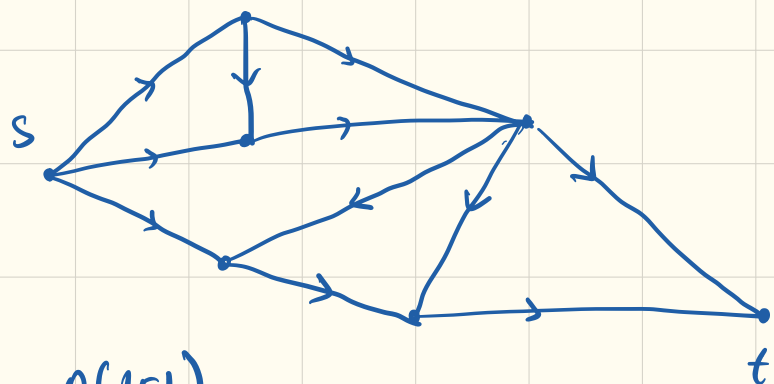


compute product when comparing

$\times$ product not sum

keep $\delta(u,v) = \dfrac{1}{r(u,v)}$ → $O(|E|)$

$r \uparrow$, $\delta \downarrow$

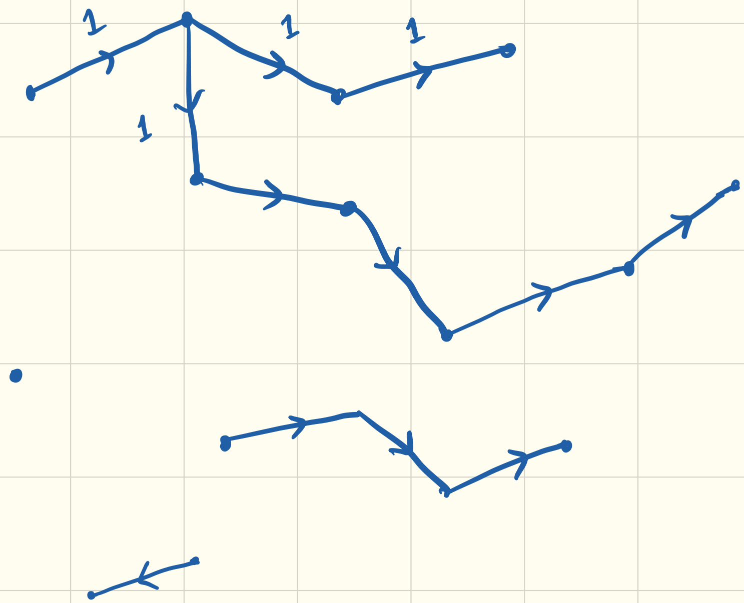


Use single source shortest path (Dijkstra's algo)

$O((|E| + |V|) \log |V|)$

$v \longrightarrow w \equiv v$ is a prerequisite for w.

$O(|E| + |V|)$



largest path

→ Find SCCs

→ Find diameters

→ Output max diameter