Network Flows – III

Case: What happens if $|C|$ is "very large".

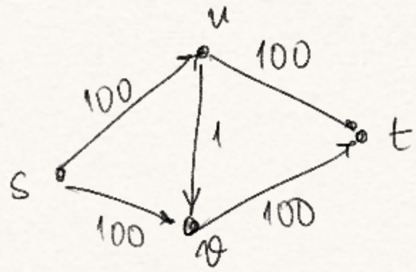
$\hookrightarrow O(|C| \cdot (m+n))$.

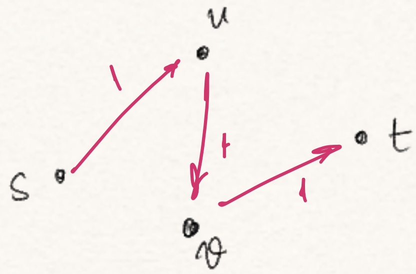
$\uparrow$

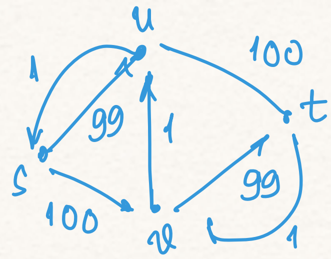
$2^{n^{\varepsilon}} \cdot (m+n)$

$\log|C| \cdot m \cdot (m+n)$

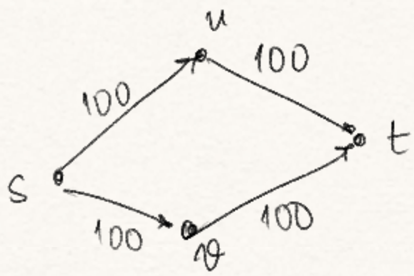
$\hookrightarrow n^{\varepsilon} \cdot m \cdot (m+n)$

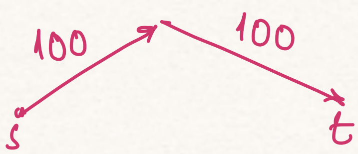


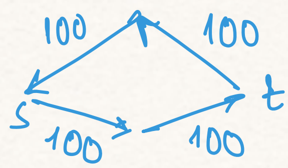


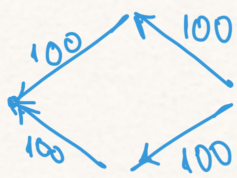


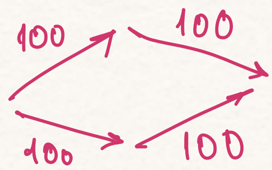
$\rightarrow$ G, C | $\underline{\underline{G(\Delta)}}$ $\leftarrow$ Graph obtained by deleting all edges of capacity $< \Delta$.

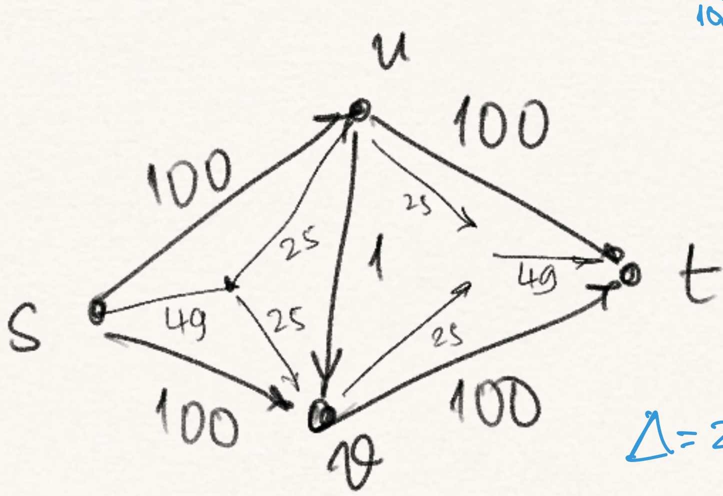




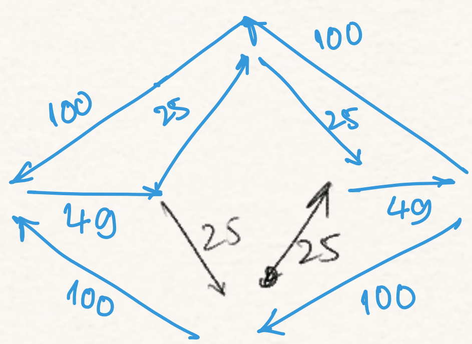


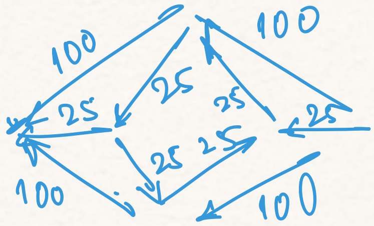


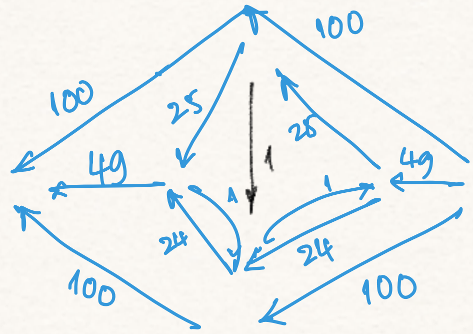




$\Delta = 25$







m

$O(\log|C| \cdot L \cdot (m+n))$

Δ-phase

L ← bound on # augmentations.

0. Set threshold to Δ

- Run algo 1 on this.
  - Each flow augmentation uses 1 BFS/DFS
- $\Delta \leftarrow \frac{\Delta}{2}$. Run step 0 again.

→ Set Δ to be the closest power of 2 just below $|C|$.

$$\Delta = \max\{2^k \mid 2^k \leq |C|\}.$$

↑ can be replaced by max capacity.

1. Construct a (residual) graph with respect to flow f w/ edges less than Δ capacity removed.

2. While there is a $s \rightsquigarrow t$ path in the updated residual graph (still with Δ threshold):

   2a. Augment the flow and construct the residual graph w/ updated flow.

3. $\Delta \leftarrow \frac{\Delta}{2}$. Go back to step 1 if $\Delta \geq 1$.

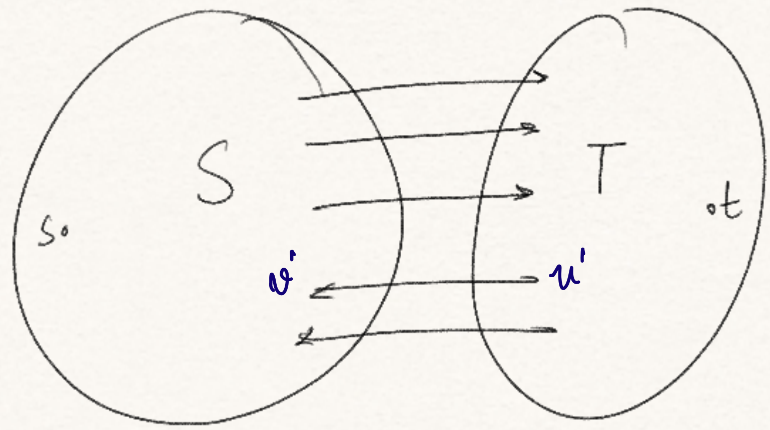
Theorem: # of iterations in Δ-phase $\leq O(m)$.

For any Δ.

Lemma: If f is the flow at the end of Δ-phase then there is a cut (S,T) in the graph of capacity at most $f + m\Delta$.

End of Δ-phase: No s-t path in the residual graph $G_f(\Delta)$. — (*)

$G_f(\Delta)$



Forward edges

$u \rightarrow v$

$c(u \rightarrow v) < f(u \rightarrow v) + \Delta$

Suppose not

$c(u \rightarrow v) \geq f(u \rightarrow v) + \Delta$

This leads to a forward s–t path w bottle neck Δ.

Contradicts *.

$(u' \rightarrow v') \in E$

$f(u' \rightarrow v') \geq \Delta$

$\Downarrow$

Backward flow/pushback of Δ can be achieved.

Contradicts *

Backward edge:

An edge where flow can be pushed back towards t from s.

$f_e < \Delta$.

Forward edges: $c_e < f_e + \Delta \rightarrow f_e > c_e - \Delta$

Backward edges: $f_e < \Delta$. $\quad -f_e > -\Delta$.

Flow across the cut $|f|$: $\sum_{\substack{u \in S \\ v \in T \\ (u,v) \in E}} f(u \rightarrow v) - \sum_{\substack{u' \in T \\ v' \in S \\ (u',v') \in E}} f(u' \rightarrow v')$

$$|f| \geq \sum_{\substack{u \in S \\ v \in T \\ (u,v) \in E}} \left[c(u \to v) - \Delta \right] - \sum_{\substack{u' \in T \\ v' \in S \\ (u',v') \in E}} \Delta .$$

$$= \sum_{\substack{u \in S \\ v \in T \\ (u,v) \in E}} c(u \to v) - \sum_{\substack{u \in S \\ v \in T \\ (u,v) \in E}} \Delta - \sum_{\substack{u' \in T \\ v' \in S \\ (u',v') \in E}} \Delta .$$

$$\geq \sum_{\substack{u \in S \\ v \in T \\ (u,v) \in E}} c(u \to v) - \sum_{(u,v) \in E} \Delta$$

$$\Rightarrow \quad \sum_{\substack{u \in S \\ v \in T}} c(u \to v) < |f| + m\Delta .$$

At the end of Δ-phase, we found a cut of capacity atmost $f + m\Delta$.

→ Any feasible flow is at most cut capacity.

$f \leftarrow$ at the end of Δ-phase.

$f' \leftarrow$ $\frac{\Delta}{2}$-phase.

$|f'| < |f| + m\Delta$.

$|f'| \geq |f| + L \cdot \frac{\Delta}{2}$. $\longrightarrow L < 2m$.

where L is the # of augmentations in the $\frac{\Delta}{2}$-phase.

$$\left.\begin{array}{l} \cancel{|f|} + L\cdot\frac{\Delta}{2} < \cancel{|f|} + m\Delta \\ \frac{L}{2} < m. \end{array}\right\} \to L < 2m.$$

Running time: $\log_2|C| \cdot 2m \cdot O(m+n)$ $\quad \} \ \Delta \to \frac{\Delta}{2} \to \frac{\Delta}{4} \dots$

$\log_k|C| \cdot k \cdot m \cdot O(m+n)$ $\quad \Delta \to \frac{\Delta}{k} \to \frac{\Delta}{k^2} \dots$

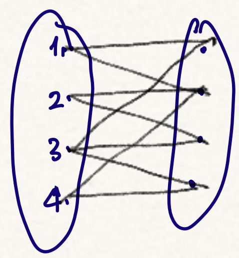
Matching (Bipartite):

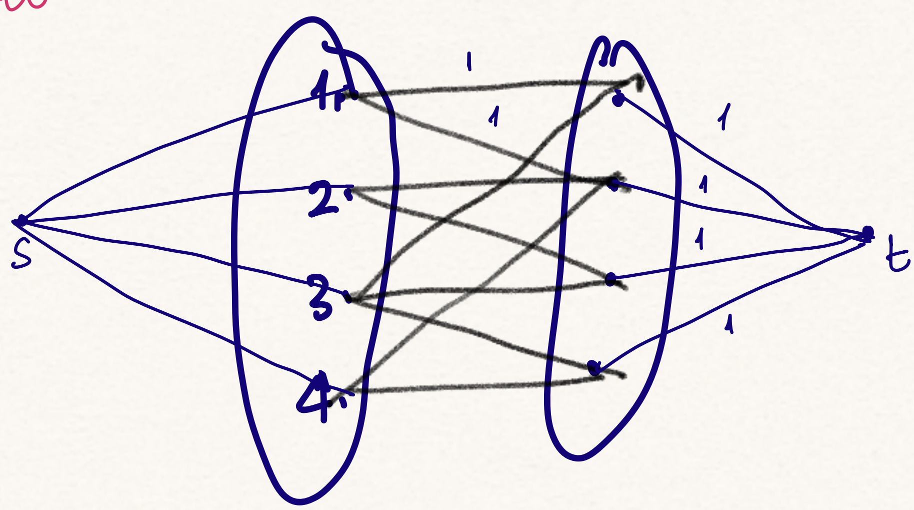
↳ Matching M is a subset of edges s.t

→ # of edges incident on any vertex is at most 1.

(Perfect matching)

↳ Every vertex has exactly 1 edge incident on it.





Give each edge a capacity 1.

Primality: Given an integer $\underline{\underline{N}}$, check if it is a prime or not!

$\log N$

For i in 1 to $\sqrt{N}$:
    check if i divides N.

if yes:
    return "Not prime".

$\sqrt{N} = N^{1/2} = 2^{\frac{\log N}{2}}$

Return "Prime".

$\rightarrow O((\log N)^{12})$ Agrawal-Kayal-Saxena.