13 March 2024

Plan.

* Max Bipartite Matching Problem
* Announcements
* Flow Reductions

Maximum Matching.

Given : Undirected Graph $G = (V, E)$

Find : maximum cardinality matching $M \subseteq E$.

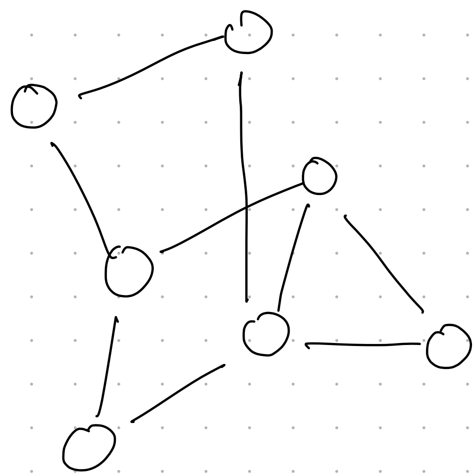
A matching is a subset of the edges M where no two edges in M share an endpoint.

Maximum Matching.

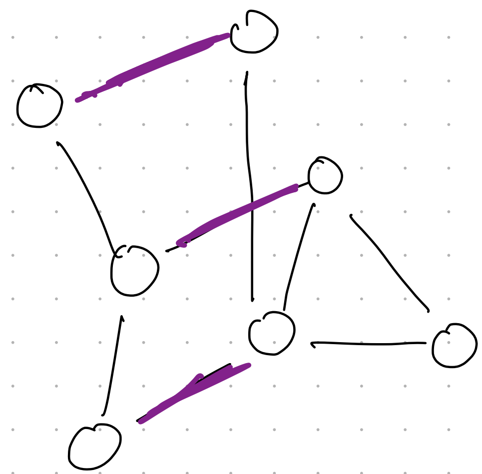
Given : Undirected Graph $G = (V, E)$

Find : maximum cardinality matching $M \subseteq E$.

$G =$



$M =$

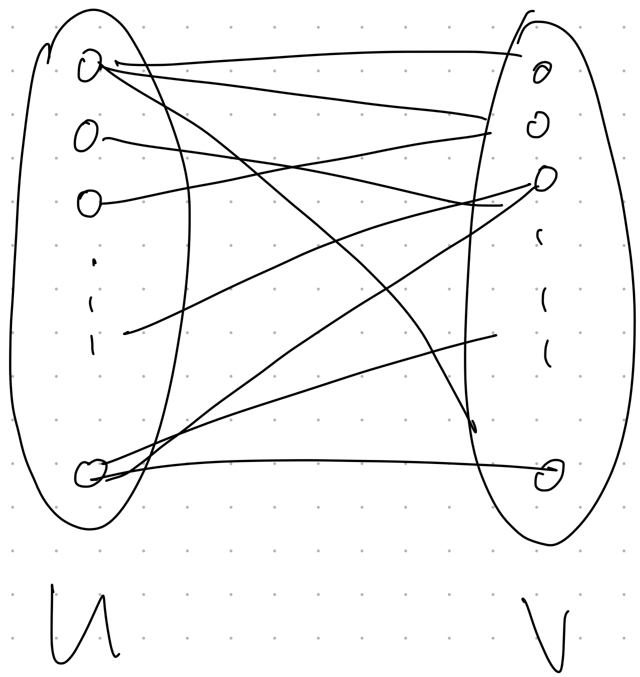


A matching is a subset of the edges M where no two edges in M share an endpoint.

Maximum Bipartite Matching.

Given: Bipartite Graph $G = (U, V, E)$

Find: Max cardinality matching

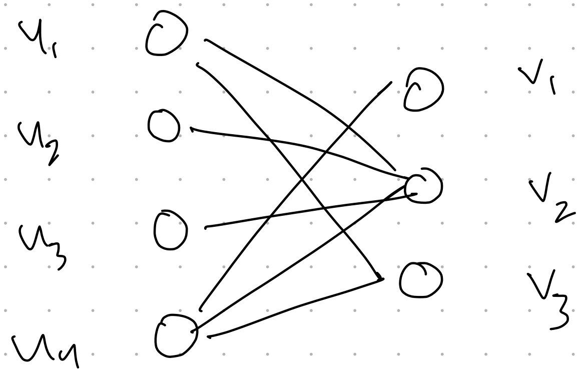


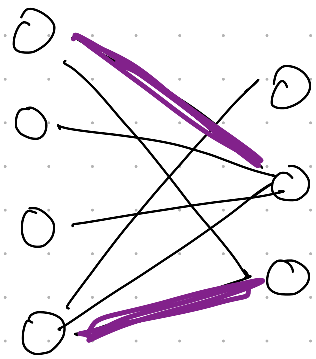
Bipartite Graph

vertices can be partitioned into U, V s.t.

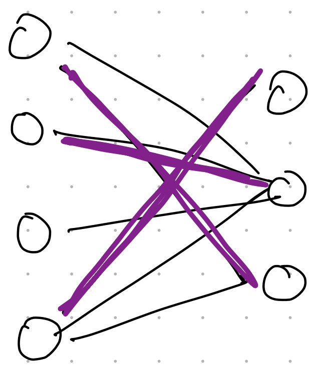
$\forall (u,v) \in E$

$u \in U$, $v \in V$



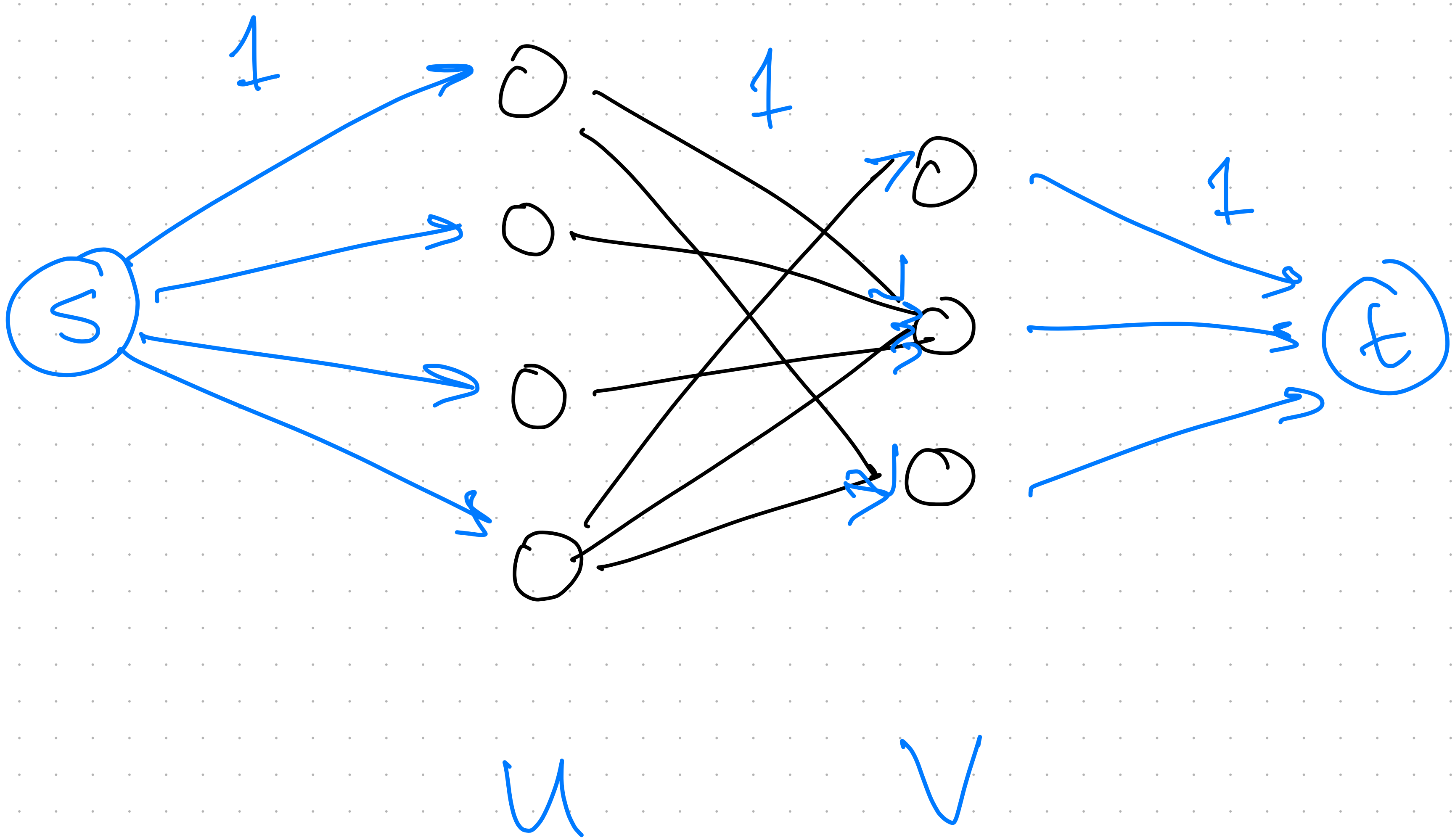


$$M = \{ (u_1, v_2), (u_4, v_3) \}$$

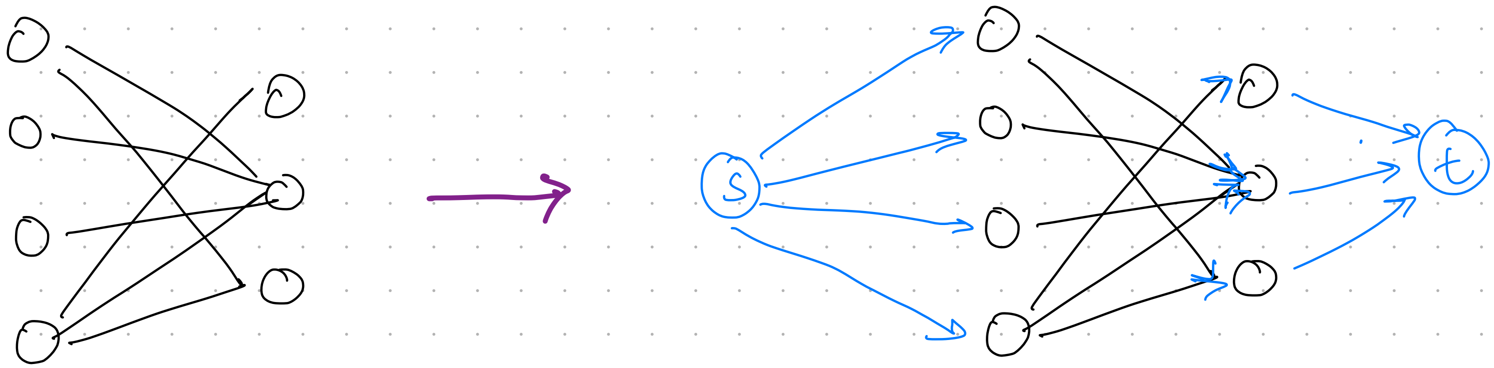


$$M' = \{ (u_1, v_3), (u_2, v_2), (u_3, v_1) \}$$

Algorithm for solving Max Bipartite Matching?



Reduction to Max Flow



Max Bipartite Match(U, V, E).

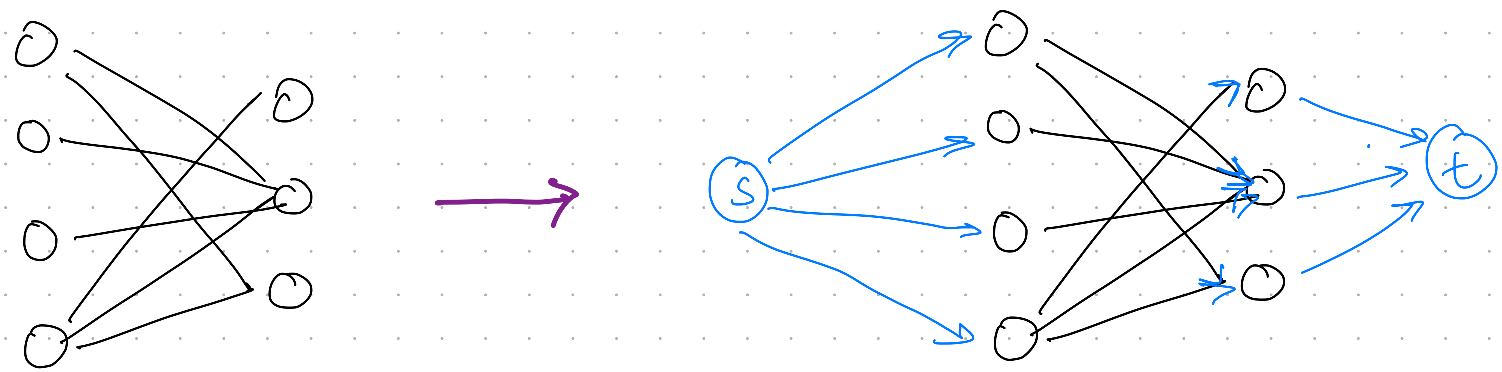
Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)

Reduction to Max Flow



Max Bipartite Match (U, V, E):

Construct directed graph G'

Vertices: $U \cup V \cup \{s, t\}$

Edges:

* for all $uv \in E$
  add $u \rightarrow v$
* for all $u \in U$
  add $s \rightarrow u$
* for all $v \in V$
  add $v \rightarrow t$

For all edges e in G'

Capacity $c_e = 1$

Return $\text{MaxFlow}(G', s, t, c)$

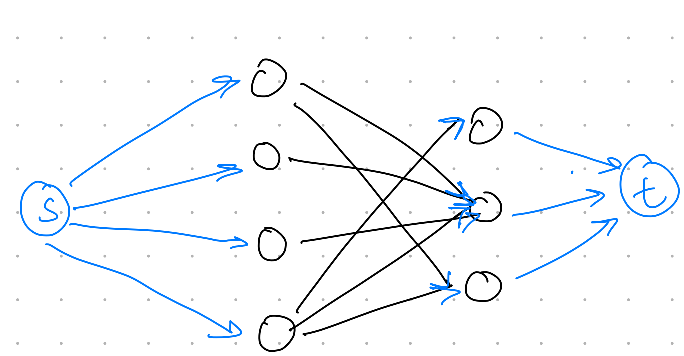
Max Bipartite Match (U, V, E) .

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return $\text{MaxFlow}(G', s, t, c)$



Claim. The max cardinality of a matching in G equals the max flow in G'

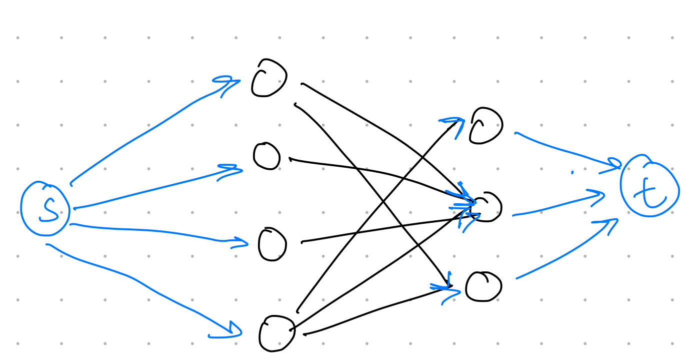
MaxBipartiteMatch(U, V, E).

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Claim. The max cardinality of a matching in G equals the max flow in G'

Reduction Proof of Correctness

$(\Rightarrow)$ If G has a matching M where $|M| = k$, then G' has a flow f where $val(f) = k$.

$(\Leftarrow)$ If G' has a flow f where $val(f) = k$, then G has a matching M where $|M| = k$.

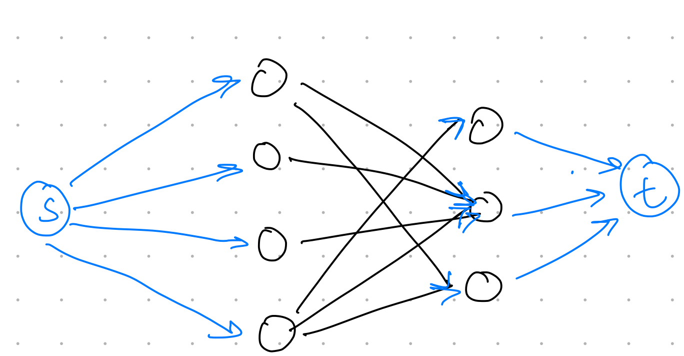
Max Bipartite Match (U, V, E):

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Claim. The max cardinality of a matching in G equals the max flow in G'

Reduction Proof of Correctness

$(\Rightarrow)$ If G has a matching M where $|M| = k$, then G' has a flow f where $\text{val}(f) = k$.

$(\Leftarrow)$ If G' has a flow f where $\text{val}(f) = k$, then G has a matching M where $|M| = k$.

($\Rightarrow$) max matching in G $\leq$ max flow in G'

Consider some matching M in G.

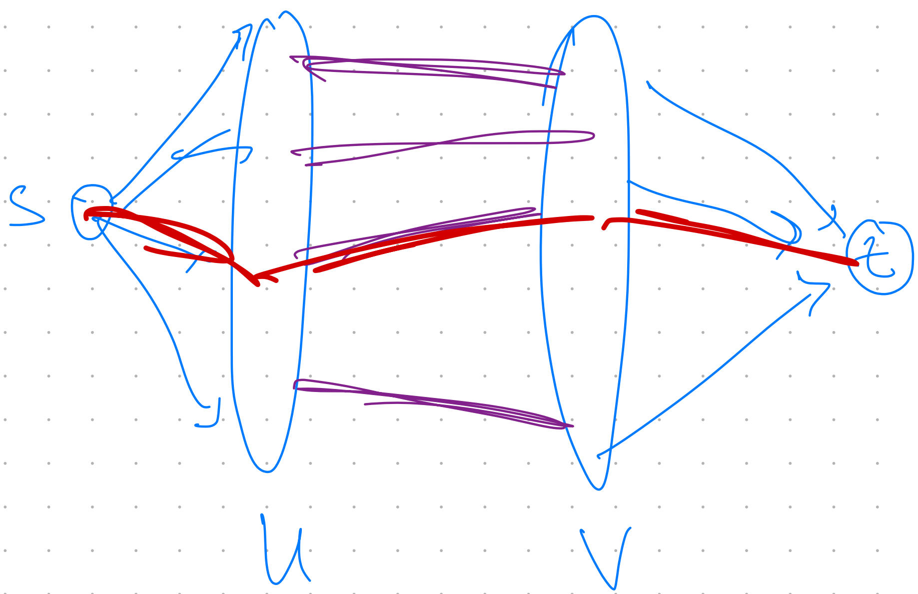
Build a flow in G' as:

- for all $e=(u,v) \in M$.

Route 1 unit of flow

$$s \xrightarrow{f_{su}=1} u \xrightarrow{f_{uv}=1} v \xrightarrow{f_{vt}=1} t$$

$|M| = k$



($\Rightarrow$) max matching in G $\leq$ max flow in G'

Consider some matching M in G.

Build a flow in G' as:

- for all $e = (u,v) \in M$.

Route 1 unit of flow

$$s \to u \to v \to t$$

$$f_{su} = 1 \qquad f_{uv} = 1 \qquad f_{vt} = 1$$

<u>Claim</u>. f is a legal flow in G'.

* <u>Capacity</u>

* <u>Conservation</u>

($\Rightarrow$) max matching in G $\leq$ max flow in G'

Consider some matching M in G.

Build a flow in G' as:

- for all $e = (u,v) \in M$.

Route 1 unit of flow

$$s \xrightarrow{f_{su}=1} u \xrightarrow{f_{uv}=1} v \xrightarrow{f_{vt}=1} t$$

Claim. f is a legal flow in G'.

By defn. of matching $u \in U$, $v \in V$ appear in M at most once.

* Capacity

Each $(s,u), (u,v), (v,t) \in G'$ cap $= 1$.

$\hookrightarrow$ Only 1 unit routed to v.

* Conservation

st-paths of flow in f are vertex-disjoint.

$\hookrightarrow$ Each path satisfies conservation, so union does as well.

($\Rightarrow$) max matching in G $\leq$ max flow in G'

Consider some matching M in G.

Build a flow in G' as:

- for all $e = (u, v) \in M$.

Route 1 unit of flow

$$s \xrightarrow{f_{su}=1} u \xrightarrow{f_{uv}=1} v \xrightarrow{f_{vt}=1} t$$

<u>Claim</u>. $val(f) = |M|$. By construction / disjointness

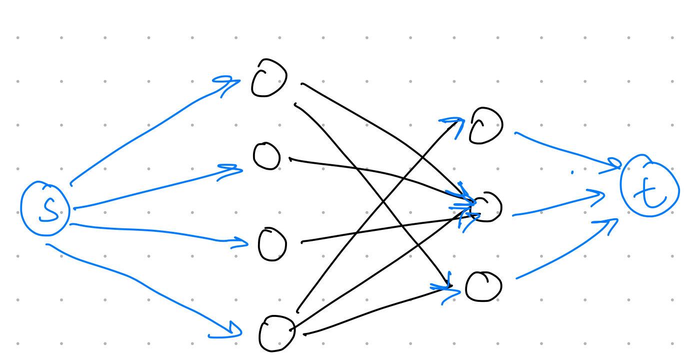
Max Bipartite Match (U, V, E).

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Claim. The max cardinality of a matching in G equals the max flow in G'

Reduction Proof of Correctness

$(\Rightarrow)$ If G has a matching M where $|M| = k$, then G' has a flow f where $val(f) = k$.

$(\Leftarrow)$ If G' has a flow f where $val(f) = k$, then G has a matching M where $|M| = k$.

($\Leftarrow$) max matching in G $\geq$ max flow in G'

Consider an integral flow f in G' (wlog)

$f_e \in \mathbb{N}$.

Build a matching $M = \{ (u,v) \in U \times V : f_{uv} = 1 \}$

Claim. M is a matching in G.

($\Leftarrow$) max matching in G $\geq$ max flow in G'

Consider an integral flow f in G' (wlog)

$\downarrow$

$f_e \in \mathbb{N}$.

Build a matching $M = \{(u,v) \in U \times V : f_{uv} = 1\}$

Claim. M is a matching in G.

* Capacity of (s,u), (v,t) is 1 for all $u \in U$, $v \in V$.

$\Longrightarrow$ For each $u \in U$, $v \in V$ at most 1 edge (u,v) has $f_{uv} = 1$.

($\Leftarrow$) max matching in G $\geq$ max flow in G'

Consider an integral flow f in G' (wlog)

$\downarrow$

$f_e \in \mathbb{N}$.

Build a matching $M = \{ (u,v) \in U \times V : f_{uv} = 1 \}$

Claim. $|M| = val(f)$

($\Leftarrow$) max matching in G $\geq$ max flow in G'

Consider an integral flow f in G' (wlog)

$\downarrow$

$f_e \in \mathbb{N}$.

Build a matching $M = \{ (u,v) \in U \times V : f_{uv} = 1 \}$

Claim. $|M| = \text{val}(f)$

- In G', every unit of $s \to t$ flow must pass through U and V
- For each unit $f_{uv} = 1$ for $u \in U$, $v \in V$

M has 1 edge (u,v).

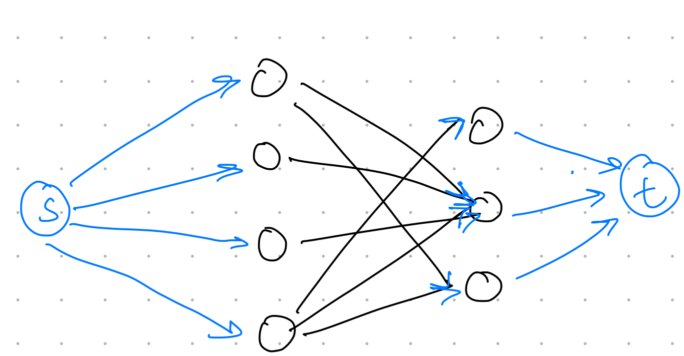
Max Bipartite Match (U, V, E).

Construct directed graph G'

For all edges e in G'

  Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Claim. The max cardinality of a matching in G equals the max flow in G'

Reduction Proof of Correctness

$(\Rightarrow)$ If G has a matching M where $|M| = k$, then G' has a flow f where $val(f) = k$.

$(\Leftarrow)$ If G' has a flow f where $val(f) = k$, then G has a matching M where $|M| = k$.

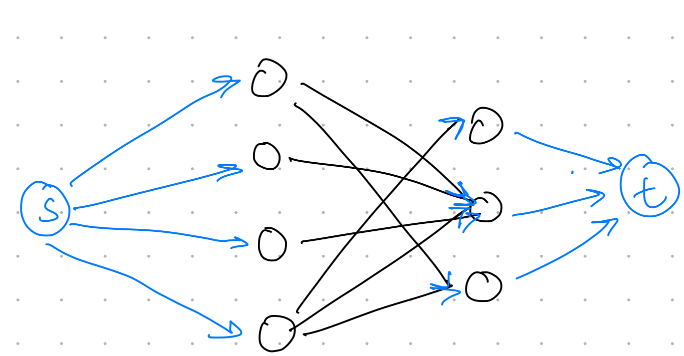
Max Bipartite Match (U, V, E) .

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Running Time

Constructing G' :

Constructing c :

Running Max Flow :

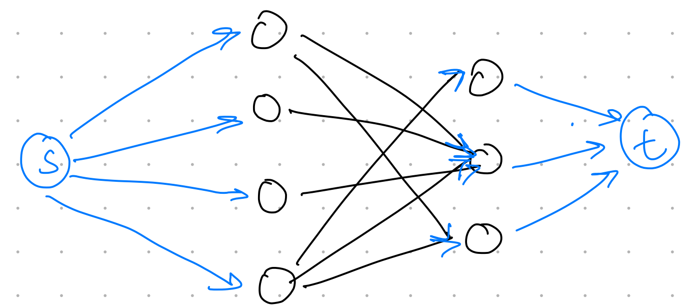
Max Bipartite Match (U, V, E).

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)



Running Time

Constructing G': $|U| + |V|$ for in/out edges per vertex

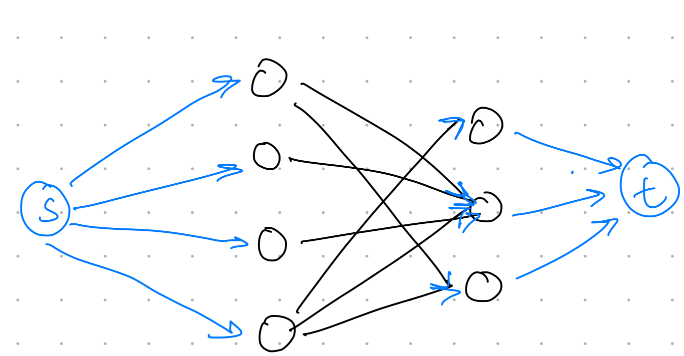
$|E|$ to direct each edge in G

Constructing c: $|E| + |U| + |V|$ for each edge in G'.

Running Max Flow:

Max Bipartite Match (U, V, E) .

Construct directed graph G'



For all edges e in G'

Capacity $c_e = 1$

Return MaxFlow(G', s, t, c)

Running Time

Constructing G' : $|U| + |V|$ for in/out edges per vertex

$|E|$ to direct each edge in G

Constructing c : $|E| + |U| + |V|$ for each edge in G'.

Running MaxFlow: $T_{MaxFlow}(|E| + |U| + |V|)$

$T_{MaxFlow}(m) \equiv$ time to solve MaxFlow in network w/ m edges.

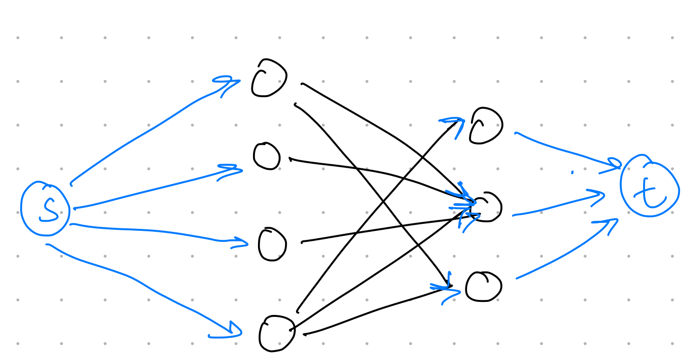
Max Bipartite Match (U, V, E).

Construct directed graph G'

For all edges e in G'

Capacity $c_e = 1$

Return $\text{MaxFlow}(G', s, t, c)$



Running Time

Constructing G' : $|U| + |V|$ for in/out edges per vertex

$|E|$ to direct each edge in G

Constructing c : $|E| + |U| + |V|$ for each edge in G'.

Running Max Flow: $T_{MaxFlow}(|E| + |U| + |V|)$

$T_{maxFlow}(m) \lesssim O(m^c)$ for any $c > 1$.

<u>Conclusion</u>. There exists an algorithm such that given a bipartite graph on $m = \Omega(|U|+|V|)$ edges, solves the maximum matching problem in $O(m + T_{MF}(m))$ time.

Announcements.

* Final Exam Scheduled

16 May 2024 7pm

No alternate exam.

(Cornell University Final Exam the only possible exception)

* HW 5 due tomorrow evening.

* Open Lecture on "Bridging Physics & Computer Science" Understanding Hard Problems.

in Rockefeller @ 7:30pm Lenka Zdeborová

Elements of a Reduction from problem P to MaxFlow

* Describe reduction R

R is an algorithm that given an instance of P, returns a flow instance
X (G, s, t, c)

Elements of a Reduction from problem P to MaxFlow

* Describe reduction R

R is an algorithm that given an instance of P, X returns a flow instance (G, s, t, c)

* Analyze Correctness

(G, s, t, c) has a flow of value K if and only if X has solution (w/ value K)

$(\Longrightarrow)$

and

$(\Longleftarrow)$

Elements of a Reduction from problem P to MaxFlow

* Describe reduction R

R is an algorithm that given an instance of P, X returns a flow instance (G, s, t, c)

* Analyze Correctness

(G, s, t, c) has a flow of value K if and only if X has solution (w/ value K)

$(\Longrightarrow)$

and

$(\Longleftarrow)$

* Analyze Running Time

$\longrightarrow$ of reduction R and solving flow instance

Baseball Elimination Problem

Teams	Wins
BOS	90
NYY	88
BAL	86
TB	91

Games

✓ (BOS, NYY) ✓

(BOS, TB) ✓

(TB, BAL) ✗

✓ (NYY, TB) ✗

BOS ?

92

NYY ?

90

TB wins, then TB > NYY

BOS wins, then BOS > NYY

Baseball Elimination Problem

Given:

* List of teams $\langle t_0, \ldots, t_k \rangle$
* Current standings $\langle w_0, \ldots, w_k \rangle$

  w_i = current # of wins by t_i

* Remaining games $\langle g_1, \ldots, g_n \rangle$

  $g_j = \underbrace{(t_i, t_k)}_{\text{Game } g_j \text{ between } t_i \text{ and } t_k.}$

<u>Question</u>: Can t_0 finish with the <u>most</u> <u>wins</u>?