

MAX FLOW

David Kauchak  
CS 1.40 – Spring 2024

1

## Admin

Assignment 10

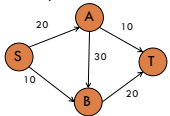
Class Thursday

2

## Flow graph/networks

Flow network

- directed, weighted graph  $(V, E)$
- positive edge weights indicating the "capacity" (generally, assume integers)
- contains a single source  $s \in V$  with no incoming edges
- contains a single sink/target  $t \in V$  with no outgoing edges
- every vertex is on a path from  $s$  to  $t$



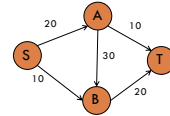
3

## Flow constraints

in-flow = out-flow for every vertex (except  $s, t$ )

flow along an edge cannot exceed the edge capacity

flows are positive



4

### Another flow problem

How much water flow can we continually send from s to t?

5

### Another flow problem

14 units

6

### Max flow problem

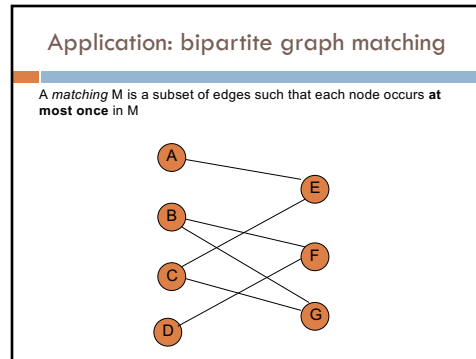
Given a flow network: *what is the maximum flow we can send from s to t that meet the flow constraints?*

7

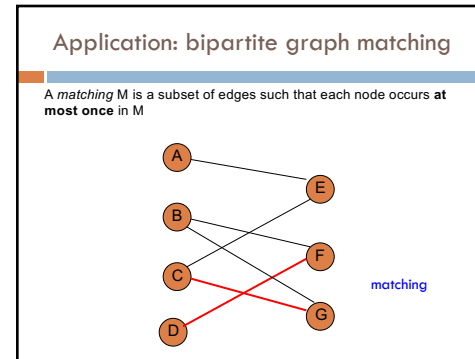
### Application: bipartite graph matching

Bipartite graph – a graph where every vertex can be partitioned into two sets X and Y such that all edges connect a vertex  $u \in X$  and a vertex  $v \in Y$

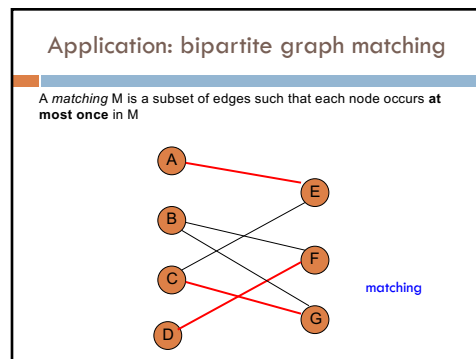
8



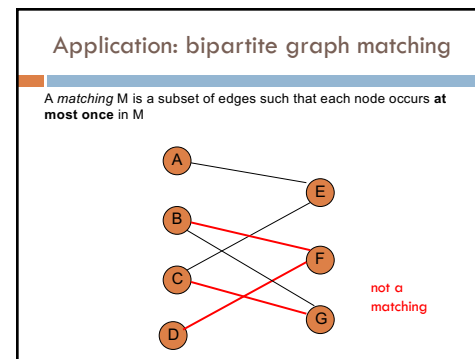
9



10



11



12

### Application: bipartite graph matching

A matching can be thought of as pairing the vertices

13

### Application: bipartite graph matching

**Bipartite matching problem:** find the *largest* matching in a bipartite graph

Where might this problem come up?

- CS department has  $n$  courses and  $m$  faculty
- Every instructor can teach some of the courses
- What course should each person teach?
- Anytime we want to match  $n$  things with  $m$ , but not all things can match

14

### Application: bipartite graph matching

Setup as a flow problem:

16

### Application: bipartite graph matching

Setup as a flow problem:

edge weights?

17

### Application: bipartite graph matching

Setup as a flow problem:

all edge weights are 1

18

### Application: bipartite graph matching

Setup as a flow problem:

after we find the flow, how do we find the matching?

19

### Application: bipartite graph matching

Setup as a flow problem:

match those nodes with flow between them

20

### Application: bipartite graph matching

**Run-time?**

**Cost to build the flow?**

- $O(E)$ 
  - each existing edge gets a capacity of 1
  - introduce  $V$  new edges (to and from  $s$  and  $t$ )
  - $V$  is  $O(E)$  (for non-degenerate bipartite matching problems)

**Max-flow calculation?**

- Basic Ford-Fulkerson:  $O(\text{max-flow} * E)$
- Edmonds-Karp:  $O(V E^2)$
- Preflow-push:  $O(V^3)$

22

### Application: bipartite graph matching

**Run-time?**

**Cost to build the flow?**

- $O(E)$ 
  - each existing edge gets a capacity of 1
  - introduce  $V$  new edges (to and from  $s$  and  $t$ )
  - $V$  is  $O(E)$  (for non-degenerate bipartite matching problems)

**Max-flow calculation?**

- Basic Ford-Fulkerson:  $O(\text{max-flow} * E)$ 
  - $\text{max-flow} = O(V)$
  - $O(V * E)$

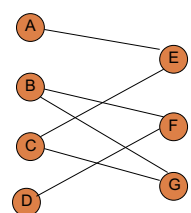
23

### Application: bipartite graph matching

**Bipartite matching problem:** find the *largest* matching in a bipartite graph

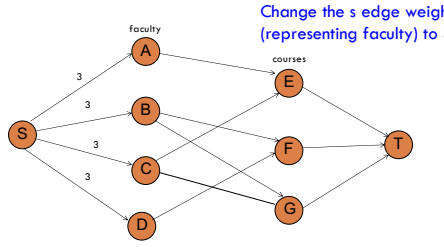
- CS department has  $n$  courses and  $m$  faculty
- Every instructor can teach *some* of the courses
- What course should each person teach?
- Each faculty can teach at most 3 courses a semester?

Change the  $s$  edge weights (representing faculty) to 3



24

### Application: bipartite graph matching



Change the  $s$  edge weights (representing faculty) to 3

All others are capacity 1

25

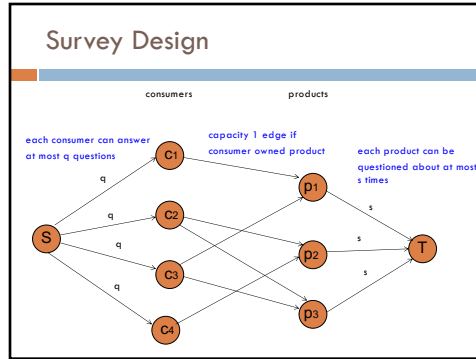
### Survey Design

Design a survey with the following requirements:

- Design survey asking  $n$  consumers about  $m$  products
- Can only survey consumer about a product if they own it
- Question consumers about at most  $q$  products
- Each product should be surveyed at most  $s$  times
- Maximize the number of surveys/questions asked

How can we do this?

26



27

### Survey design

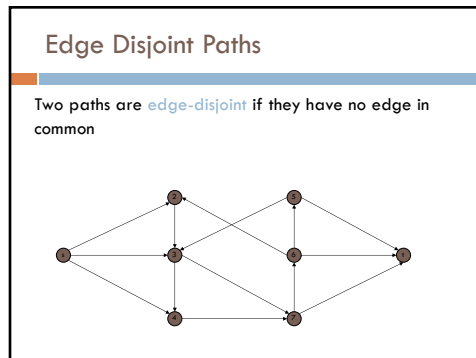
Is it correct?

- Each of the comments above the flow graph match the problem constraints
- max-flow finds the maximum matching, given the problem constraints

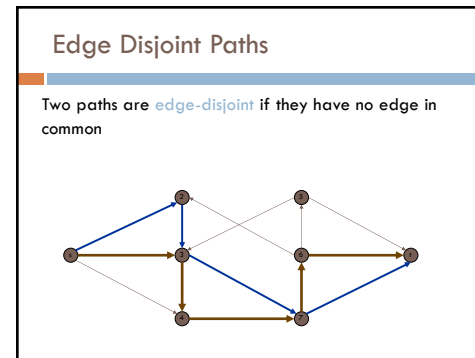
What is the run-time?

- Basic Ford-Fulkerson:  $O(\text{max-flow} * E)$
- Edmonds-Karp:  $O(V E^2)$
- Preflow-push:  $O(V^3)$

28



29



30

### Edge Disjoint Paths Problem

Given a directed graph  $G = (V, E)$  and two nodes  $s$  and  $t$ , find the max number of edge-disjoint paths from  $s$  to  $t$

Why might this be useful?

31

### Edge Disjoint Paths Problem

Given a directed graph  $G = (V, E)$  and two nodes  $s$  and  $t$ , find the max number of edge-disjoint paths from  $s$  to  $t$

Why might this be useful?

- ▣ edges are unique resources (e.g. communications, transportation, etc.)
- ▣ how many concurrent (non-conflicting) paths do we have from  $s$  to  $t$

32

### Edge Disjoint Paths

Algorithm ideas?

33

### Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge

What does the max flow represent?  
Why?

34



### Edge Disjoint Paths

Max flow formulation: assign unit capacity to every edge

- max-flow = maximum number of disjoint paths
- correctness:
  - each edge can have at most flow = 1, so can only be traversed once
  - therefore, each unit out of s represents a separate path to t

35

### Max-flow variations

What if we have multiple sources and multiple sinks (e.g. the USSR train problem has multiple sinks)?

36

### Max-flow variations

Create a new source and sink and connect up with infinite capacities...

37

### Max-flow variations

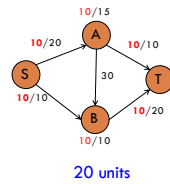
Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex

What is the max-flow now?

38

### Max-flow variations

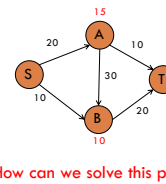
Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex



39

### Max-flow variations

Vertex capacities: in addition to having edge capacities we can also restrict the amount of flow through each vertex

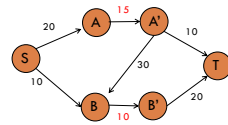


How can we solve this problem?

40

### Max-flow variations

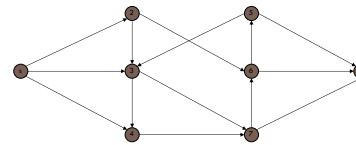
- For each vertex  $v$ 
  - create a new node  $v'$
  - create an edge with the vertex capacity from  $v$  to  $v'$
  - move all outgoing edges from  $v$  to  $v'$



41

### More problems: maximum independent path

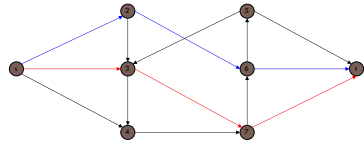
Two paths are *independent* if they have no *vertices* in common



45

### More problems: maximum independent path

Two paths are **independent** if they have no vertices in common

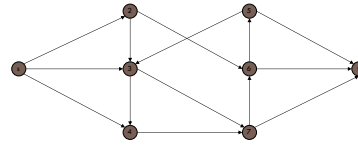


46

### More problems: maximum independent path

Find the maximum number of independent paths

Ideas?

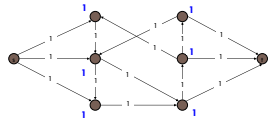


47

### maximum independent path

Max flow formulation:

- assign unit capacity to every edge (though any value would work)
- assign unit capacity to every vertex



Same idea as the maximum edge-disjoint paths,  
but now we also constrain the vertices

48