Objectives

- Network Flow Wrap Up
  - Application: Airline Scheduling
- Algorithms Retrospective
- Computational intractability

Review

- What is a flow network?
- What is our usual goal given a flow network?
  - How do we reach that goal?
- What is the Ford-Fulkerson algorithm?
  - What is its runtime?
- What is the min-cut?
  - How does it relate to the max flow?
- How is network flow useful?

Analyzing Augmenting Path Algorithm

```
Ford-Fulkerson(G, s, t, c)
O(m)   foreach e ∈ E  f(e) = 0  # initially no flow
O(m)   Gf = residual graph
Find path: O(m); Iterations: O(F) iterations, where F = max flow
       while there exists augmenting path P
O(m)      f = Augment(f, c, P)     # change the flow
O(m)      update Gf              # build a new residual graph

       return f
```

Total: O(Fm)

```
Augment(f, c, P)
O(n)  b = bottleneck(P) # edge on P with least capacity
O(n)  foreach e ∈ P
O(1)     if (e ∈ E) f(e) = f(e) + b  # forward edge, ↑ flow
O(1)     else       f(eR) = f(e) - b  # forward edge, ↓ flow
      return f
```

Total: O(n) → O(m), since n ≤ 2m

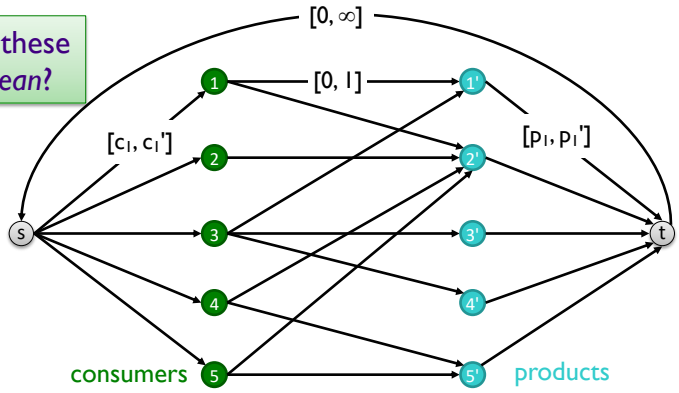
Review

- What was our solution (so far?) to the survey design problem?
- What is our process in solving problems using network flow?

Survey Design Algorithm

- Formulate as a circulation problem with lower bounds
  - Include an edge (i, j) if customer i owns product j

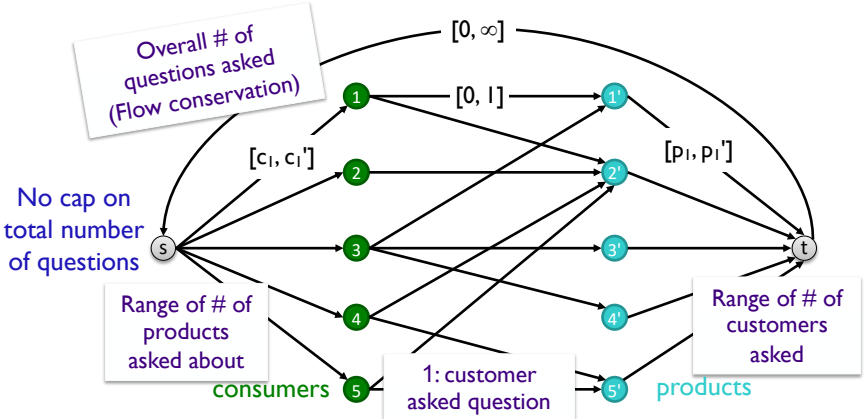
What do these edges *mean*?



Survey Design Algorit

- Formulate as a circulation
  bounds
  - Include an edge (i, j) if cust

Alternative bounds on t→s?
How do we know if we can create a survey?
What is the survey?
How many solutions are there to this problem?



Survey Solution - Analysis

- How do we know that the solution found is correct/feasible/optimal?
- How do we know that we found all solutions?
- Analyze run time
  - Creating model
  - FF algorithm

Survey Solution

- If there exists a feasible, integer flow solution to this graph, we can create the survey
  - And, vice versa: If we can create the survey, there is a feasible, integer flow solution to this graph
- Customer *i* will be surveyed about product *j* iff the edge (*i,j*) carries a unit of flow
  - Flow from s→customer: number of questions the customer is asked
  - Flow from product→t: number of questions asked about the product

Survey Solution Analysis

- Build the model: O(nk)
  - Create nodes for each consumer (n) and product (*k*)
  - Create nodes s and t
  - Create directed edge from s→customer
  - Create directed edge from product→t
  - Create directed edge customers to products
- Run FF: O(Fm), where F is flow, m is number of edges
  - But flow can be infinity, so… O(∞nk)?
    - Better to have a tighter bound on flow
      - e.g., sum of all questions to customers, …

Review: Network Flow Solutions

1. Model problem as a flow network
   - Describe what nodes, edges, and capacity represent
   - Describe what flow represents and how that maps to your solution
   - Run Ford-Fulkerson algorithm
     - Map back to original problem
2. Prove that the solution found is correct/feasible/optimal
3. Prove that you find all solutions
4. Analyze running time
   - Creating model
   - FF algorithm

7.9 AIRLINE SCHEDULING

Airline Scheduling

- **Scheduling goal**: efficient in terms of equipment usage, crew allocation, customer satisfaction, …
- **Our simplified problem:**
  - Flight segment: origin & destination airport, departure & arrival time
  - Use a plane for two flight segments (i, j) if
    - i's destination == j's origin & enough time to perform maintenance on plane **OR**
    - Add a flight segment in between that gets plane to j's origin with adequate time in between

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

What is a valid use of one plane for > 1 segment?

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

What is a valid use of one plane for > 1 segment?

1 → 3 → 6

Problem Statement

- A flight j is *reachable* from flight i if it is possible to use the same plane for flight j as flight i

Goal: Determine if it's possible to serve all *m* flights using at most *k* planes

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

Could we schedule all flights from with only 2 planes?

Scheduling Planes

- Maintenance time: 1 hour

Number	Origin	Departure	Destination	Arrival
1	Boston	6 a.m.	DC	7 a.m.
2	Philadelphia	7 a.m.	Pittsburgh	8 a.m.
3	DC	8 a.m.	LAX	11 a.m.
4	Philadelphia	11 a.m.	San Francisco	2 p.m.
5	San Francisco	2:15 p.m.	Seattle	3:15 p.m.
6	Las Vegas	5 p.m.	Seattle	6 p.m.

Yes.
Plane A: 1 → 3 → 5
Plane B: 2 → 4 → 6

Problem Statement

- A flight j is *reachable* from flight i if it is possible to use the same plane for flight j as flight i

Goal: Determine if it's possible to serve all *m* flights using at most *k* planes

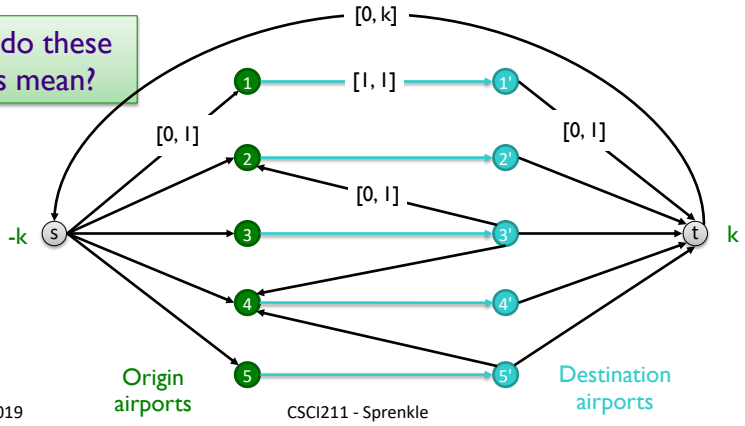
Ideas about our solution/approach?

Airline Scheduling Algorithm

- Flow: airplanes; Nodes: airports
- Find a feasible circulation

[lowerbound, upperbound]

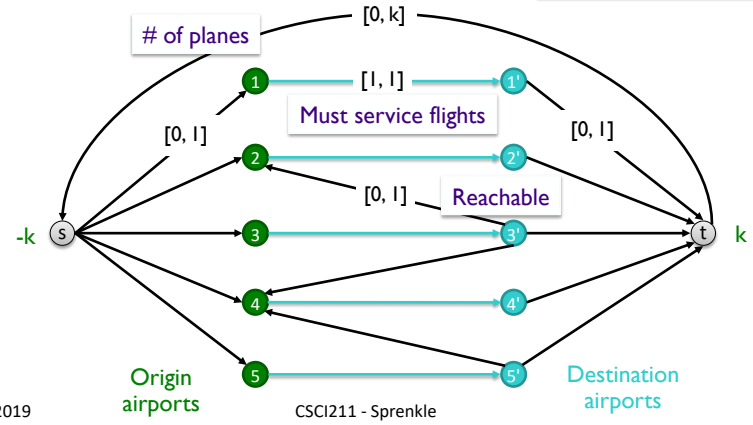
What do these edges mean?



Airline Scheduling Algorithm

- Flow: airplanes; Nodes: airports
- Find a feasible circulation

How do we know if we have a solution? How do we get the solution?



Scheduling Solution

Goal: Determine if it's possible to serve all *m* flights using at most *k* planes

- Model
  - Flow: airplanes
  - Nodes: airports
- Use FF algorithm to generate flow
  - If feasible flow → feasible circulation
- Construct schedules by following edges with flow from *s* to origin airports
  - The flow represents the schedule of one plane

Analyzing Schedule Solution

Goal: Determine if it's possible to serve all *m* flights using at most *k* planes

- Cost of building the graph?
- Cost of generating flow?

Analyzing Schedule Solution

Goal: Determine if it's possible to serve all *m* flights using at most *k* planes

- Cost of building the graph: $O(m^2)$
  - Each flight (m)
    - Two nodes, one directed edge, with capacity [1,1]
  - Reuse plane edges (m^2) with flow [0,1]
  - Edge from source to each flight (m)
  - Edge from each flight to sink (m)
- Cost of generating flow: $O(k\ m^2)$
  - FF is O(Fm), where F is flow, m is number of edges
    - For our problem, F is *k* and m is m^2

Total: $O(k\ m^2)$

Network Flow Solutions

1. Model problem as a flow network
   - Describe what nodes, edges, and capacity represent
   - Describe what flow represents and how that maps to your solution
   - Run Ford-Fulkerson algorithm
2. Prove that the solution found is correct/feasible/optimal
3. Prove that you find all solutions
4. Analyze running time
   - Creating model
   - FF algorithm

Looking Ahead

- PS 9 due Friday
- Course Evaluations Distributed
  - If 60% of you complete the survey, 1% extra credit added to your problem set grade
  - For each additional 10% who complete the survey, additional 1% added to your problem set grade
- Final Exam handed out on Friday