CSE 421 Section 7

Max Flow / Min Cut

Administrivia

Announcements & Reminders

- Midterm Exam
    - Not everyone has taken the exam so please do not discuss it with others until we indicate that it is OK to do so.

- HW6
    - Due next week, Wednesday 11/15

Ford-Fulkerson Algorithm

Finding the Max-Flow / Min-Cut

We use the Ford-Fulkerson algorithm to find Max-Flow / Min-Cut.

Key Ideas:

- Keep searching through the residual graph to find a path from s to t that we can send more flow.
- Keep updating the residual graph to track how much flow we can still push through and how much flow we can potentially reroute.
- When we can no longer reach t in the residual graph, we can't send any more flow, so the algorithm terminates!

Residual Graph

The residual graph indicates how much flow can still go along an edge, and how much flow we could potentially reroute back from an edge.

Key ideas:

- the sum of the residual edges between any two nodes should be equal to the value of the edge between them in the original graph
- The residual edge pointing in the direction of the original edge should have a value equal to the amount of flow that could still pass through that edge
- The residual edge pointing in the opposite direction of the original edge should have a value equal to the amount of flow you have currently sent down that edge

Ford-Fulkerson (formally)

While (flow is not maximum)

Run BFS in residual graph starting from s

Record predecessors to find an s,t-path

Iterate through path, finding c minimum residual capacity on path

Add c to every edge on path in flow

Update residual graph

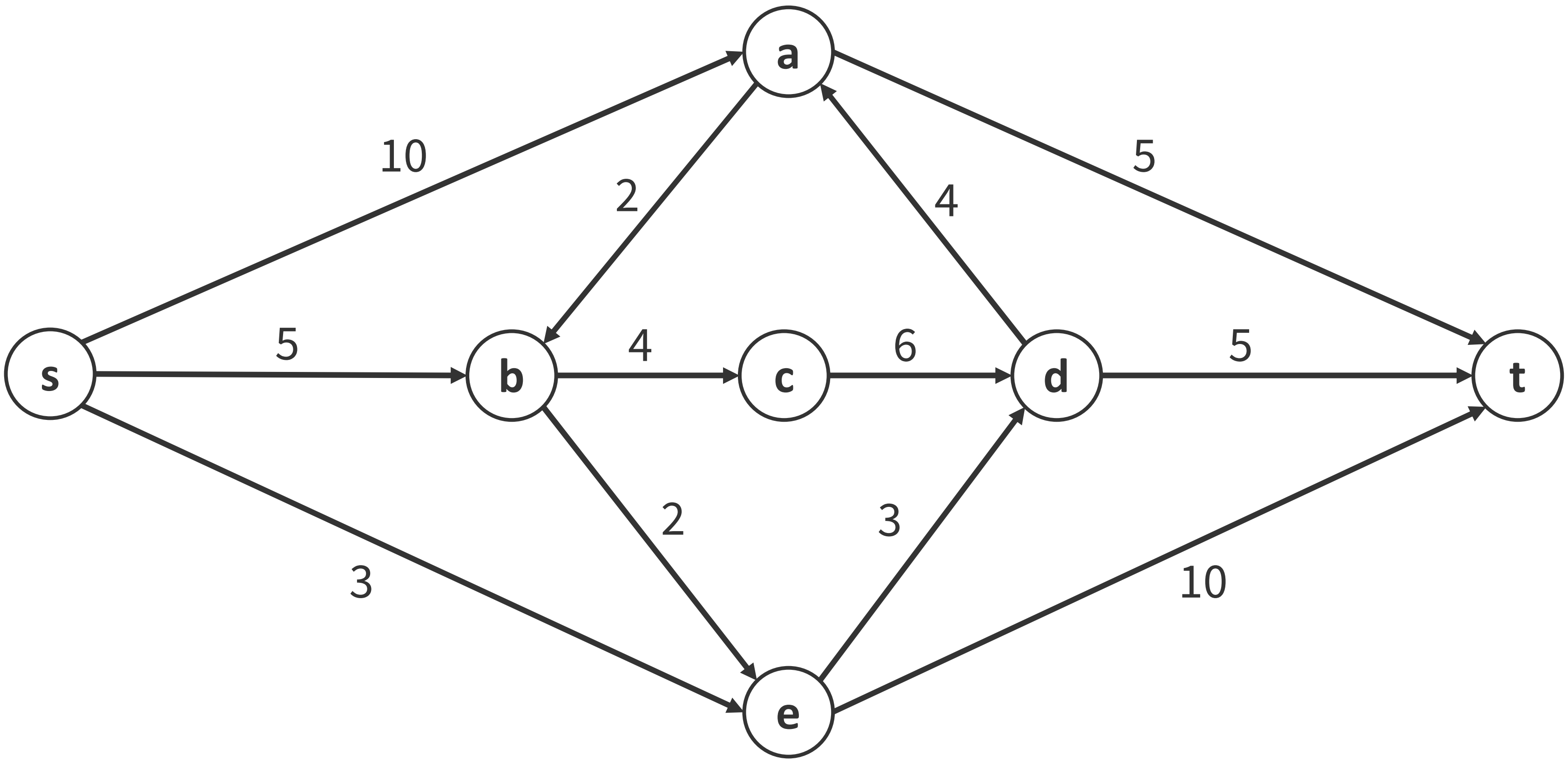
1. Go With the Flow

Problem 1 – Go With the Flow

Using Ford-Fulkerson, find the maximum $s - t$ flow in the graph G below, the corresponding residual graph, and list out the corresponding minimum cut.

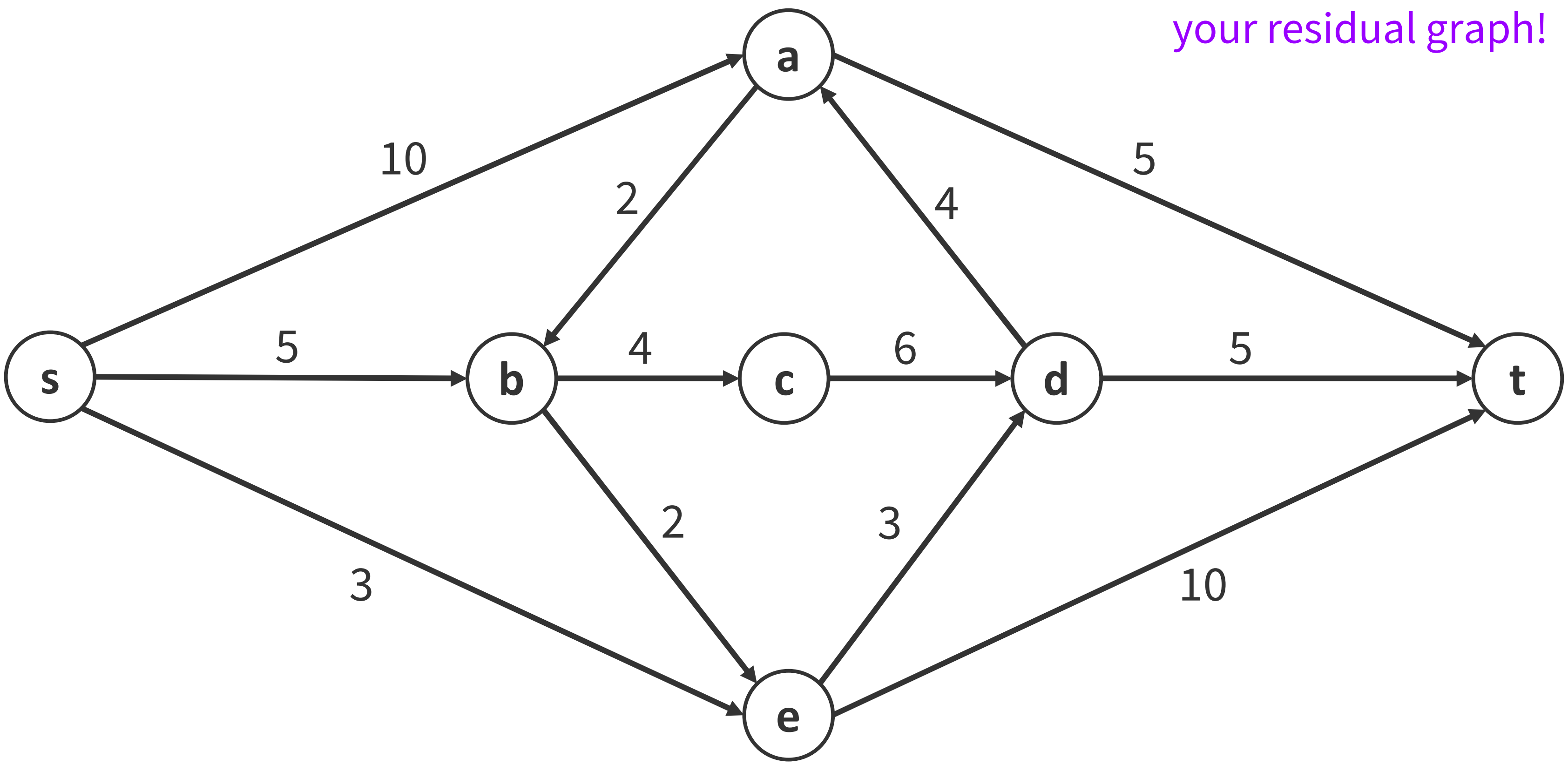
Work through this problem with the people around you, and then we'll go over it together!

Problem 1 – Go With the Flow



Problem 1 – Go With the Flow

Start by making a copy of the original graph to be your residual graph!



Max-Flow / Min-Cut Tricks

Max-Flow / Min-Cut

We can use the concepts of Max-Flow / Min-Cut and the Ford-Fulkerson algorithm to solve a wide variety of problems. Since we already have an algorithm, we can just call it like a library function.

Most of the difficulty comes in taking a problem and turning it into a good graph so that max-flow / min-cut gives us the solution we are actually looking for. So how can we do it?

The Strategy

1. Read the Problem Carefully
2. Make a Basic Model
3. Brainstorm: How can you fix the graph?
4. Correctness and Running Time

The Tricks

We have three tricks that can be really helpful in converting a problem into a good form for max-flow or min-cut. Sometimes you only need one, but sometimes you can use them in a combination. There are other things you might need to do in a given problem, but these are three very common tricks to try:

- Add “dummy vertices” for source or sink
- Split vertices to add vertex capacity
- Use infinite weight for edges that shouldn’t be considered for max-flow or min-cut

2. You’re Not a Dummy…

Problem 2 – You’re Not a Dummy…

You have three overfilled reservoirs and two underfilled reservoirs. You want to (as quickly as possible) move a total of 10,000 gallons of water from the overfilled reservoirs to the underfilled ones. You do not care how much comes from each of the three individual reservoirs (as long as the total is 10,000 gallons) nor how much arrives at each of the underfilled ones (again, as long as the total is 10,000 gallons). You have a map of (one-way) pipes connecting the reservoirs (in the form of a directed graph); each pipe has a maximum capacity in gallons per minute. **You wish to find the way to route the water and the amount of time that will be required.**

Problem 2.1 – Read the Problem Carefully

Answer the usual quick-check questions:

- Are there any technical terms in the problem you don't know? Are there any words that look like normal words, but are actually technical terms?

- What is the input type?

- What is the output type?

Work through this problem with the people around you, and then we'll go over it together!

Problem 2.1 – Read the Problem Carefully

Answer the usual quick-check questions:

- Are there any technical terms in the problem you don't know? Are there any words that look like normal words, but are actually technical terms?

- What is the input type?

- What is the output type?

Problem 2.2 – Make a Basic Model

This sounds like a flow problem. From what you know so far, what would the flow model be? What parts of the problem have you represented successfully? What is still missing?

Work through this problem with the people around you, and then we'll go over it together!

Problem 2.2 – Make a Basic Model

This sounds like a flow problem. From what you know so far, what would the flow model be? What parts of the problem have you represented successfully? What is still missing?

Problem 2.3 – Brainstorm: How can You Fix It?

Find a clever trick to represent the missing piece. The goal here is to do a reduction. By the end of this step, you should have a “standard” flow problem.

Work through this problem with the people around you, and then we’ll go over it together!

Problem 2.3 – Brainstorm: How can You Fix It?

Find a clever trick to represent the missing piece. The goal here is to do a reduction. By the end of this step, you should have a “standard” flow problem.

Problem 2.4 – Correctness and Run Time

Explain why your algorithm is correct. For flow problems, the proof is usually just explaining how you've represented each part of the problem and relying on the correctness of the flow algorithm.

Work through this problem with the people around you, and then we'll go over it together!

Problem 2.4 – Correctness and Run Time

Explain why your algorithm is correct. For flow problems, the proof is usually just explaining how you've represented each part of the problem and relying on the correctness of the flow algorithm.

3. Split Personality

Problem 3 – Split Personality

You have been given a map of the water cleaning system for the city of Seattle. Water enters from a marked vertex and flows through pipes (directed edges with specified capacities), through processing facilities, and back out to nature (marked as a specified sink vertex). The processing facilities are vertices in your graph. As the facilities process the water, they **also** have maximum capacities, which may be less than the total capacity entering or leaving the vertex. You wish to **find the amount of water that can flow through this network** while **respecting both the facility and pipe capacities**.

Problem 3.1 – Read the Problem Carefully

Answer the usual quick-check questions:

- Are there any technical terms in the problem you don’t know? Are there any words that look like normal words, but are actually technical terms?
- What is the input type?
- What is the output type?

Work through this problem with the people around you, and then we’ll go over it together!

Problem 3.1 – Read the Problem Carefully

Answer the usual quick-check questions:

- Are there any technical terms in the problem you don’t know? Are there any words that look like normal words, but are actually technical terms?
- What is the input type?
- What is the output type?

Problem 3.2 – Make a Basic Model

This sounds like a flow problem. From what you know so far, what would the flow model be? What parts of the problem have you represented successfully? What is still missing?

Work through this problem with the people around you, and then we'll go over it together!

Problem 3.2 – Make a Basic Model

This sounds like a flow problem. From what you know so far, what would the flow model be? What parts of the problem have you represented successfully? What is still missing?

Problem 3.3 – Brainstorm: How can You Fix It?

Find a clever trick to represent the missing piece. The goal here is to do a reduction. By the end of this step, you should have a “standard” flow problem.

Work through this problem with the people around you, and then we’ll go over it together!

Problem 3.3 – Brainstorm: How can You Fix It?

Find a clever trick to represent the missing piece. The goal here is to do a reduction. By the end of this step, you should have a “standard” flow problem.

Problem 3.4 – Correctness and Run Time

Explain why your algorithm is correct. For flow problems, the proof is usually just explaining how you've represented each part of the problem and relying on the correctness of the flow algorithm.

Work through this problem with the people around you, and then we'll go over it together!

Problem 3.4 – Correctness and Run Time

Explain why your algorithm is correct. For flow problems, the proof is usually just explaining how you’ve represented each part of the problem and relying on the correctness of the flow algorithm.

That’s All, Folks!

Thanks for coming to section this week!
Any questions?