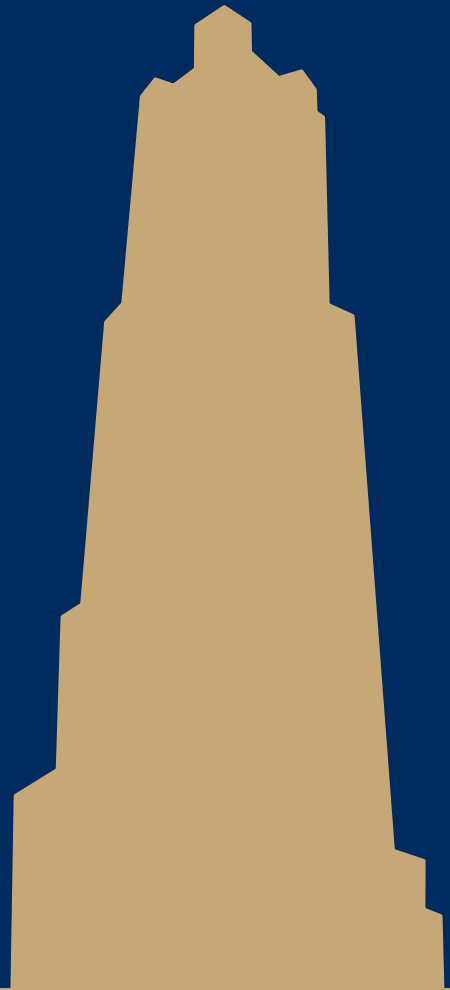


CS/COE 1501

www.cs.pitt.edu/~lipschultz/cs1501/

Final Review



Questions/Comments about final exam

- How much time do we have to complete the exam?
 - 1.5 hours (6pm - 7:30pm)
- Is it going to be the same length as the midterm or is it going to be longer?
 - Approximately the same length
- Can we expect the final exam to be the same format as the other exams, or will it be different?
 - It will be the same format as the midterm.
 - Mostly have short answer questions
 - Some multiple choice (or true/false, matching, etc)

Questions/Comments about final exam

- Is it cumulative or just the material we covered after the midterm?
 - It is mostly material not covered by the midterm
- It would be most helpful to know what each problem will relate to in terms of the topic. Example: Problem 1 will be union find, problem 2 will be P vs. NP, etc. With material this hard, it makes it more fair for us to study since not only is there a lot of information, but the information is extremely difficult.
 - Part of the challenge is identifying which topic a problem belongs to. Outside of this class, you won't be told "This graph problem is actually a dynamic programming problem."

What do you want the review session to look like?

- 86%: Lecture-style (instructor goes through topics submitted through a survey and talks about them)
- 14%: Question and answer (students ask questions during class and instructor answers them)

Topics to Review

- 38.8%: P vs. NP
- 20.8%: Cryptography
- 9.7%: Network Flow
- 9.1%: Dynamic Programming
- 6.4%: Weighted Graphs
- 6.2%: Union Find
- 6.0%: More Math (Exponentiation & GCD)
- 3.0%: Integer Multiplication

P vs. NP

- Why unsolvable problems matter to us
- I would like to know the difference between P and NP problems
- Which problems should be solved by P algorithms and which by NP algorithms
- Which statement is correct:
 - P and NP algs solve the same set of problems, or
 - all P problems can be solved by NP algorithms, and a subset of NP problems can be solved by P algorithms

P vs. NP

- I would like to go through another example determining whether or not a problem is P or NP.
- I especially don't fully understand how to prove NP completeness for a problem.
- I would also like to see a simple proof that shows how to prove if a problem is NP complete or not.
- Examples

NP-Completeness Proof Example

Is the 0/1 Knapsack Problem problem NP-Complete?

P vs. NP

- I am wondering if we will have to actually prove NP completeness on the final or if we just have to list the steps of proving NP completeness.

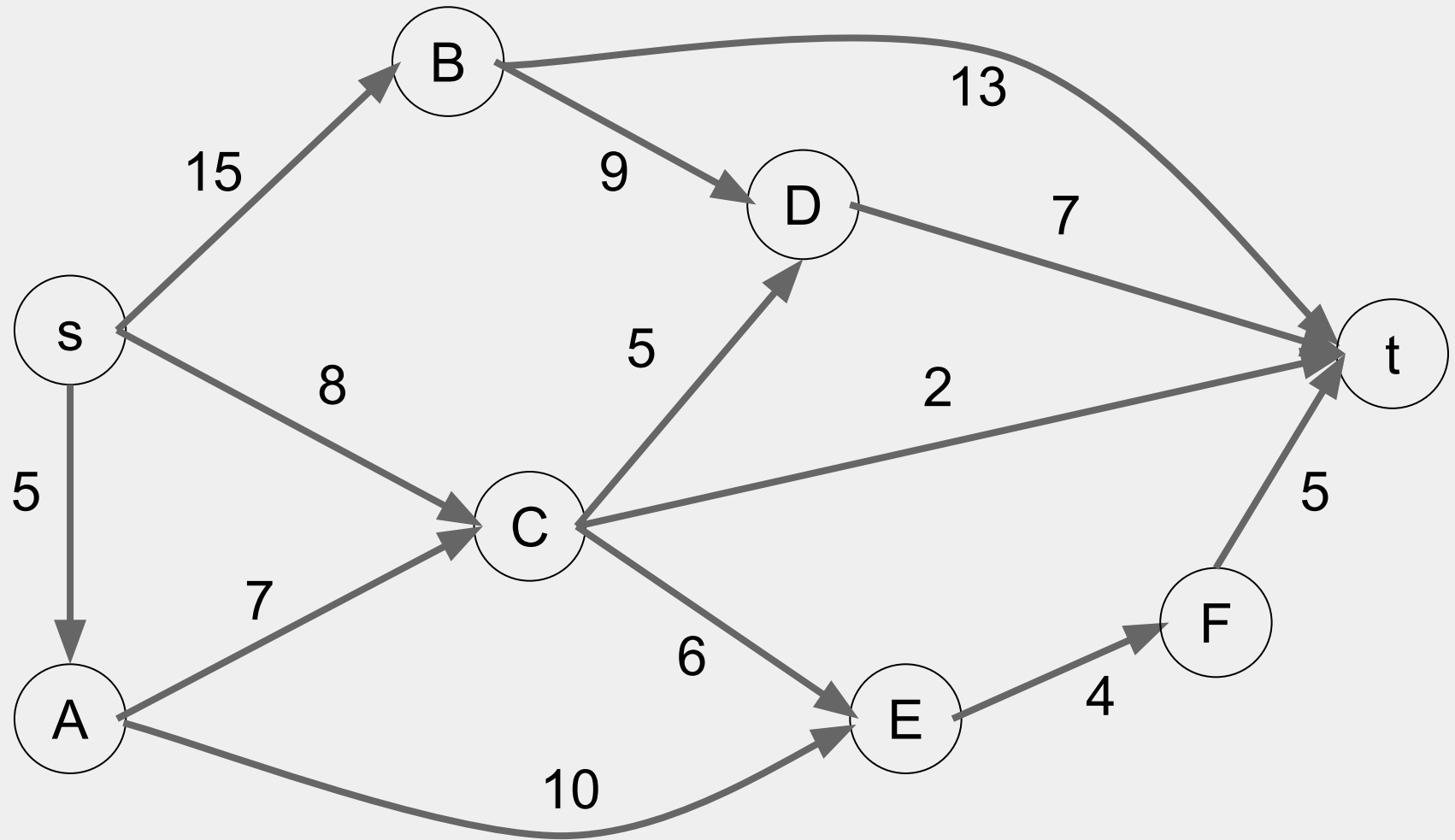
Cryptography

- Best method for encrypting and decrypting (symmetric, AES, RSA), modular arithmetic, inverses (encrypting is the inverse of decrypting? how)

Network Flow

- I don't understand pretty much anything about network flow. If we could go over just about anything about it, it would help.
- I would like to see a trace of each algorithm that was covered during the network flow lecture.
- I just want to be sure I have this down. An example with min cut would be good and also using back flow.

Network Flow Example



Dynamic Programming

- Difference between dynamic and greedy problems. More specifically, the difference between a subproblem and a choice (property 2 appears to be the same as property 1 to me)

Requirements for Greedy Algorithms to Produce Optimal Results

- Problems must have two properties to guarantee optimal results:
 - Optimal substructure
 - Optimal solution to a subproblem leads to an optimal solution to the overall problem
 - The greedy choice property
 - Globally optimal solutions can be assembled from locally optimal choices

Dynamic Programming

- 0/1 knapsack problem matrix (how it works)

Weighted Graphs

- I'd like to review examples of the implementations for algorithms involving weighted graphs, especially Dijkstra's algorithm and Kruskal's algorithm.

Union Find

- I have trouble understanding how Kruskal's algorithm works for union find.

More Math

- There was a lot of complex proof style math done in this section, it would be nice to see another example or maybe an outline of what we should know for this section.

Integer Multiplication

- I would like to see how the improved grade school algorithm is obtained, and I would also like to see some examples of applying the master theorem to different algorithms.