CSCE 411: Design and Analysis of Algorithms

Final Exam Review

1 Test format and instructions

- Some multiple choice questions, some long answer questions
- You do not need to prove something unless you are explicitly asked to prove something
- You may be asked to explicitly prove something
- Scratch paper will be provided
- No calculators or electronic devices. You will not need them. If you have to do any computations, they will be basic enough to do by hand.
- You will put your name on the front page, and put your initials at the top of every other page.

The basic format will be mostly the same as Tests 1 and 2, just longer. More multiple choice, and more free answer.

- 15 multiple choice questions
- 4 free answer questions

2 What have we covered since Test 2 that you should know?

General concepts for complexity

- Definitions of: P, NP, NP-complete, NP-hard, and co-NP
- Understand basic open questions, e.g., P vs. NP
- What is required to prove something is in each of these problem classes
- What a reduction is, and what it means for it to be a polynomial time reduction
- How to use reductions to prove something is in P or in NP. (Make sure you understand how to beware of pitfalls in proving reductions!)
- There are some problems you should know are NP-complete: clique, graph coloring, vertex cover, clique cover, SAT, 3SAT, MAXCUT, subset sum, Hamiltonian path
- There are some problems you should know are in P: 2SAT, and pretty much all of the problems we covered before Test 2

Reductions and problem we especially considered

- Graph clique and relationship with vertex cover
- SAT, especially 2SAT and 3SAT

Approximation algorithms and optimization

- You should know how to write a linear program and the dual linear program for an optimization problem
- You should know that there exist polynomial-time algorithms for solving linear programs
- Weak duality and strong duality theorems

3 What should I prioritize in my studying?

What to study, roughly in the order I'd suggest you prioritize:

1. The practice homework associated with the last week of lectures on NP-hardness reductions
2. Practice problems at the end of this lecture
3. The overview in these lecture notes
4. Homeworks associated with last unit
5. Study guides for Tests 1 and 2
6. Everything else we covered during the course (lectures and homework)

I suggest covering topics using a breadth first search rather than a depth first search. (Why preparation using depth first search can backfire: https://xkcd.com/761/).

4 Practice Problems

1. The decision problem of finding whether G has a clique cover of size k is denoted

$$\textsc{CliqueCover}(G,k) = \{\langle G,k\rangle : \text{ determine if } G \text{ is has a clique cover of size } k\}$$

   (a) Prove that CliqueCover is in NP.

   (b) Prove that a graph $G = (V, E)$ has a k-coloring if and only if the complement graph $\bar{G} = (V, \bar{E})$ has a clique cover of size k.

   (c) Explain why the above fact implies that CliqueCover and GraphColoring are either both in P, or are both NP-complete.

2. Bob is trying to prove that 2SAT can be reduced in polynomial time to the Clique problem. Do we already know whether this reduction is possible, or do we know for sure that it is impossible? Or is it an open question whether it is or isn't possible? If he succeeds, what (if anything) will this tell us about the computational complexity of these problems?

3. Steve is trying to prove that every instance of 3SAT can be reduced in polynomial time to an instance of 2SAT. Answer the same questions that you answered for problem 2.

4. Alice is trying to come up with a polynomial time reduction from Clique to Max-Cut. Answer the same questions that you answered for problem 2.

5. Is the graph coloring problem NP-complete for general k? Is it still NP-complete when $k = 2$? Is it still NP-complete if $k = 3$?

6. What is the difference between the class of NP-complete problems and the class of NP-hard problems? Are these classes exactly the same? If they are not exactly the same, explain how these two classes of problems differ.

7. Susie says she came up with a polynomial time reduction from Subset-Sum to a new problem called VertexHopping. She explains to her friend Doug that this proves that the VertexHopping problem is NP-hard. Doug claims that this doesn't prove NP-hardness, because a problem is only NP-hard if *every* problem in NP can be reduced to it, whereas Susie only proved that one specific problem in NP (Subset-Sum) can be reduced to Vertex Hopping. Is Doug right? If not, explain why he is wrong.

8. Let MaxFlow(G, s, t, k) represent the *decision version* of the maximum s-t flow problem where you are given a weighted directed graph with nodes s and t and the problem asks whether there is an s-t flow function that sends k units of flow from s to t. Prove that this problem is in co-NP.

9. A factory produces two drinks, A and B. Each liter of product A requires 2 hours of labor and 1 unit of raw material. Each liter of product B requires 1 hour of labor and 2 units of raw material. The factory has a total of 100 hours of labor and 80 units of raw material available. The profit from each unit of product A is \$40, and from each unit of product B is \$30. The factory wants to maximize its total profit, assuming the amount of each drink produced does not need to be an integer. Write the linear program associated with this optimization problem. Write the dual linear program.