**CSCI 2300 — Introduction to Algorithms**
**Exam 2 Prep and Sample Questions (document version 1.0)**

- Exam 2 is scheduled for 5:00-9:59PM EDT on Thursday 4/23
- Exam 2 is open book(s), open notes; given that you are working remotely, you may use any and all of the posted course materials, including all previous questions and answers posted in the Discussion Forum.
- Make-up exams are given only with an official excused absence (`http://bit.ly/rpiabsence`)
- Exam 2 covers everything from after Exam 1 through Thursday 4/16 lecture, which therefore includes DPV sections 2.1-2.4, 5.1-5.2, 5.4, 6.1-6.4, 6.6-6.7, and 7.1-7.2 (i.e., Homeworks 3, 4, 5, and 6, as well as Labs 4, 5, and 6)
- To prepare for the exam, focus on the warm-up and homework/lab problems, as well as the problems discussed and/or proposed in the corresponding lectures slides
- Key topics are greedy algorithms, divide-and-conquer, dynamic programming, linear programming, and network flow
- All work must be your own; do not even think of copying or communicating with others
- Be as concise as you can in your answers; long answers are difficult to grade

## Sample problems

Work on the problems below as practice problems as you prepare for the exam. **Also work on the "Warm-up problems" and graded problems from the homeworks and labs noted above.**

Feel free to post your solutions in the Discussion Forum (except for graded Homework 6 problems); and reply to posts if you agree or disagree with the proposed approaches/solutions.

Some selected solutions will be posted on Wednesday 4/22.

1. Show that if an undirected graph with $n$ vertices has $k$ connected components then it has at least $n - k$ edges.
2. Using a Huffman encoding of $n$ symbols with frequencies $f_1, f_2, \cdots, f_n$, what is the longest a codeword could possibly be?
3. Give a linear-time greedy algorithm that takes as input a tree and determines whether it has what is called a *perfect matching*, i.e., a set of edges that touches each vertex exactly once.

4. For a set of $n$ variables $x_1, x_2, \cdots, x_n$, we are given $q$ *equality* constraints of the form $x_i = x_j$ and $p$ *disequality* constraints of the form $x_i \neq x_j$. Let $m = q + p$.

   Is it possible to satisfy all given constraints? More specifically, give an efficient greedy algorithm that models the problem as a graph and takes as input $m$ constraints over $n$ variables, providing as output whether the constraints can be satisfied or not. What is the runtime of your algorithm?

   As an example, the following set of constraints cannot be satisfied:

   $$x_1 = x_2, x_2 = x_3, x_3 = x_4, x_1 \neq x_4$$

5. Given two sorted lists of size $m$ and size $n$, describe a greedy algorithm that computes the $k$th smallest element in the union of the two lists. Use a divide-and-conquer approach and make sure your algorithm runs in $O(\log m + \log n)$ time.

6. DPV Problem 2.18

7. DPV Problem 6.7

8. DPV Problem 6.9

9. DPV Problem 6.11

10. DPV Problem 7.5

11. DPV Problem 7.6

12. DPV Problem 7.7