CSE 101 WINTER 2018 - FINAL EXAM ADVICE

Basic Preparation steps

- **[Don't ignore nutrition. In evening finals, a lot of students tend to zone out and be very inefficient during the last part of the exam. Quite often, this is a nutrition / coffee-instead-of-food issue. There will be some emergency granola bars etc. at the exam, but don't rely on this as something to sustain you through 3 hours.]**
- **Understand the likely scope of the exam**
  - 6 questions, 60 points total; "proportional" within limits of discreteness (questions, lectures are discrete)
  - **Half** the points (**~3 questions'** worth) are "available by short-answer or mechanical execution of algorithms" (this is standard in exams). Which suggests:
    - TTK ?
    - Coverage of Weeks 1-4 (maybe 15-ish % of points on the test)
    - Ford-Fulkerson for maximum flow; reductions of matching → flow → LP are examples of "mechanical"
    - Writing down an LP for a simple "scenario" (bowls/mugs, cargo weight (HW5 Q6) for example) is considered "mechanical"
  - **~2 Algorithm design questions**
    - Greed is certain
    - DP is certain
  - **~1 NP-completeness reduction**
    - **Mentioned on Piazza: @1029**
    - **Please review the structure of proofs that some Problem A is NP-complete.**
    - **(1) We prove that Problem A is in NP.**
    - **(2) We prove that some known NP-complete Problem B is polynomial-time reducible to Problem A.**
    - **……… *(This being said, any question on the final exam would be much simpler - in a similar vein to the HP/HC or TSP / TSP-E, problem 0/0' etc. examples that you've seen.)***
- **Make your own "mental map" of key concepts, topics, examples seen in lecture, HW and discussion**
  - Know what has been covered == what is considered "fair game"
  - Greed: proving optimality
  - DP: "patterns" (1-D prefixes, 1-D intervals, 2-D pairs of prefixes, …)

- Examples and variants: knapsack variants; common sequences in strings; shortest paths (B-F, F-W == transitive closure) and number of paths in graphs; ...
- Similarities / analogues:
  - matrix chain product, optimal parenthesization, polygon triangulation, HW4 Q3 …. All of these are O(n^3) DP examples that follow very similar patterns
  -
- It is **IMPORTANT** that you fully know the contents of discussion sections. All material from weekly discussion sections aid/enhance knowledge of the class material, and are in scope of the final exam.
- Remember to read the **solutions** of each homework, even if you scored a full score in each question. However, we DO NOT expect you to write solutions at the level of detail as in our solutions.

Tips for excelling on the day of exam

- Have a strategy on what questions you will attempt first. You don't have to solve the questions in the order they appear.
- ***Read instructions on first page carefully.***
- ***You should have an idea of how many questions there are before attempting the paper.***
  - ***(It's known that there are 5 main questions.)***
  - ***(Slide shown at start of exam will remind you to read over the questions at the start of the exam. And, FWIW, we feel that Questions 1 and 2 (total = 30 points) will require less than half the exam time. We expect that you will have the majority of exam time to work on Questions 3, 4 and 5 (total = 30 points).***
  - ***Question 1 is T/F. Only T/F/blank answers are needed. No justifications.***
    - ***Grading is +1 for correct, -1 for incorrect, 0 for left blank -- so that there is no benefit from random guessing.***
  - ***Question 2 is Flow/LP.***
  - ***Question 3 is Greed.***
  - ***Question 4 is DP.***
  - ***Question 5 is NP / Reduction.***
- Approach the exam with confidence. If an answer does not strike you immediately, do not fret or lose confidence. Read the question again, and see if you missed some given information. See if you are using all the information given in the problem.
- ***If you are stuck on a question for a while, move onto the next question.*** Coming back later gives you time to work on other questions, and also look at this question with a second perspective in the end.
- **Problem types**:

  - You should not be spending too much time on **True/False** questions unless they carry a significant proportion of the score.
  - Devote enough time to **algorithm design questions** as they carry maximum weight *(? - may be outdated)*. Below are tips for each algorithmic design pattern, if one appears in the exam:
    - On a **greedy algorithmic design** question, remember that proof of optimality is *usually* of higher weightage than the algorithm itself. *outdated?* Form a mental model of how your proof should shape up and write it down neatly and concisely. *Other guidelines from homework apply.*
    - On a **dynamic programming algorithmic design** question, remember to write down the sub problem definition and the recursive formulation (mathematical, not in English) clearly, without which we will be unable to grade your answer. *Other guidelines from homework apply.*
    - On a **Divide & Conquer algorithmic design** question, the recurrence relation is expected as well. *Other guidelines from homework apply.* *deprecated/outdated: algorithm design questions are Greed and DP*
    - On a **network flow** question where a network is to be constructed, make sure you describe your intuitions/reasoning for the graph construction and its edges and capacities. Draw the network and label/outline everything. Do not leave anything ambiguous. Do as directed for mechanical questions. *Other guidelines from homework apply.*
    - On **NP** questions, do as directed.
- **Writing answers**:
  - Remember to write down all the components required carefully, concisely and neatly.
  - We suggest using a pen, but if you are using a pencil, make sure you use a pencil with enough lead darkness to be able to read on a scanned version.
  - If you are scribbling a part of the answer elsewhere, make sure you indicate this clearly at the original answer space providing the grader enough information to correctly grade your answer. We suggest not dividing answer into parts (i.e. one part in one page, and the other part in scratch), *unless you have to.*
- Remember to approach the paper *greedily! :)* Maximize your score with the right approach and attitude!