CSOR W4231.001 — Midterm Exam Instructions

1. Please write your name and UNI at the top of every page you scan.
2. The exam is open textbook (CLRS), open lecture slides, lecture notes and homework solutions. No other resources may be consulted (e.g., Internet, other textbooks, etc.). You must adhere to your signed honor pledge and the CS department academic honesty policies. Failure to do so will result in receiving a 0 in the exam and further disciplinary actions.
3. You should read all the problems as soon as possible as they might not appear in order of difficulty to you.
4. Please keep your answers clear and concise. You will be graded not only on the correctness of your answers, but also on the clarity with which you express them.
5. For all algorithms you design, you must describe them clearly in English, argue correctness even if you do so briefly, and analyze the running time. You should also follow any other specific instructions in the problems (for example, analyze space if asked to do so). Pseudocode is optional unless explicitly stated otherwise.
   - If you give a Greedy algorithm, you must give a full proof of correctness to get full marks.
6. If you give a Dynamic Programming algorithm, you must
   - Clearly define the subproblems.
   - Give the recurrence and explain it briefly in English. (You do not need to give the inductive proof of correctness.)
   - State boundary conditions.
   - State running time.
   - State space complexity.
   - Give the order to fill in subproblems if the DP table has at least two dimensions.
7. You can use any algorithm that we covered in class or the homework assignments by simply referring to it and specifying its input, then stating its running time.
8. Your hand-writing should be very clear and your scan should be high quality. We cannot grade your answer if we have difficulty reading it.
9. **You must follow the instructions at EdStem post #2 to submit your exam. You must submit your exam to Gradescope to receive a non-zero grade. There will be no exception to this rule.**You must submit a single pdf with your scanned handwritten answers to Gradescope under assignment "Midterm Exam" within 90 minutes of the time you start the exam.
10. **No modifications to your submission will be permitted.** You should double check that all your answers appear in the pdf you will upload to Gradescope.
11. You should sign the honor pledge before submitting your exam. If you have not already submitted your signed honor pledge, you should handwrite, date, sign and include it in the pdf with your exam answers that you will submit to Gradescope. We will not grade exams for which we have not received the signed honor pledge.

Duration: 75 mins to complete the exam + 15 additional mins to scan and upload your answers.
Total: 110 points

GOOD LUCK!

1. Problem 1 (4 points)

The following statements are True or False.

Write the correct answer as **True** or **False** on your sheet. **You must write the full word: if you write T or F, you will not receive any credit.** No explanation required. All log are $\log_2$.

i. The asymptotic runtime of Mergesort does not change if, instead of generating 2 subproblems each of size $n/2$, we generate and solve recursively 3 subproblems each of size $n/3$.

ii. $n^{0.1} = \Omega((\log n)^{10})$

2. Problem 2 (24 points)

The following statements are either **True** or **False**. State what they are (**True** or **False**) and prove your answers by providing a short explanation or a suitable counter-example.

(a) (12 points) The following is a correct algorithm for finding the shortest path from node s to node t in a directed graph with some negative weight edges but no negative cycles:

Add a large constant to each edge weight so that all the edge weights become positive. Then run Dijkstra's algorithm from s and return the shortest path found to t.

(b) (12 points) The solution to the recurrence

$$T(n) = T(n-1) + T(n-2), \text{ for } n \geq 2$$
$$T(0) = T(1) = 1$$

is $T(n) = \omega(2^n)$.

3. Problem 3 (25 points)

You are given a directed weighted graph $G = (V, E, w)$ with positive edge weights, where $n = |V|$, $m = |E|$, $w_e > 0$ is the weight of edge e.

Design an algorithm that, on input $G = (V, E, w)$ returns the minimum weight of any cycle in the graph, or "no" if G is acyclic. Your algorithm should take time at most $O(n^3)$.

The weight of a cycle is defined as the sum of the weights of the edges on the cycle. In symbols, if $C = \{v_1, v_2, \ldots, v_{k-1}, v_k = v_1\}$ is a cycle, then $w(C) = \sum_{i=1}^{k-1} w_{v_i v_{i+1}}$.

4. Problem 4 (30 points)

There are $n > 1$ cards laid out in a line; the i-th card from the left has value $v_i > 0$.

At each step, you collect two adjacent cards. If the values of the two cards you collected are u, v, then you make a *profit* $u + v$. Then the two cards are removed and replaced with a new card of value $u+v$ (the new card is placed at the gap created by the removal of the two cards).

The process terminates when there is exactly 1 card left on the table.

Your goal is to maximize your *total profit* upon termination of the process. The total profit is defined as the sum of the *profits* you made at every step until the process terminated. Design an efficient algorithm that, on input $\{v_1, \ldots, v_n\}$, returns the maximum *total profit*.

5. Problem 5 (27 points)

(a) (10 points) We use Huffman's algorithm to obtain an encoding of alphabet $\mathcal{A} = \{a, b, c, d, e\}$ with probabilities $\mathcal{P} = \{p_a, p_b, p_c, p_d, p_e\}$. In each of the following cases, you must either

- Give an example of probabilities p_a, p_b, p_c, p_d, p_e that would yield the specified codeword length sequence (note that in this case, **you must give actual probabilities**); Or
- Explain why the code cannot possibly be obtained no matter what the probabilities are.

The codeword length sequences follow:

i. Sequence of codeword lengths for code C_1: $\{1, 2, 3, 4, 4\}$

ii. Sequence of codeword lengths for code C_2: $\{1, 3, 3, 3, 3\}$

(b) (17 points) Suppose that our alphabet consists of the first 6 letters of the English alphabet A through F. I have a file where A appears 6 times, B appears 15 times, C appears 20 times, D appears 20 times, E appears 15 times and F appears 6 times.

Draw the Huffman tree for this input on your paper, give the Huffman code and determine the average codeword length of the Huffman code.