

CS/ECE 374 A (Spring 2022)

Conflict Midterm Exam 2

Instructions:

- Date/Time: April 12 Tuesday 7:00pm–9:30pm Central Time.
- Except for your one double-sided handwritten $8.5'' \times 11''$ cheat sheet, exams are **closed-everything**. In particular: No medically unnecessary electronic devices are allowed in exams, including smart watches and headphones/earbuds.
- You will write your solutions on paper, scan your solutions using a scanning app, convert your scans to PDF, and upload the PDF to Gradescope. (You are **not** allowed to write your solutions on tablets, unlike in some past semesters.) Gradescope will only accept PDF submissions. Please make sure your solution to each numbered problem starts on a new page of your submitted PDF file. Please do not scan your cheat sheets or scratch paper.
- You have **two hours (i.e., 120 minutes)** to write your solutions. We are providing 30 minutes at the end of each exam period for students to scan and upload their exams, and to deal with any unexpected technical difficulties that may arise. Gradescope will stop accepting submissions for each midterm precisely at 9:30pm. We will not accept submissions after the Gradescope deadline. **All work on the exam must stop 30 minutes before the Gradescope deadline, i.e., at 9:00pm.**
- Please make sure that your scans are clear and easy to read. We very strongly recommend that you write with black ink on unlined white paper. Submissions consisting of raw photos or low-quality scans will be penalized.
- If you are ready to scan your solutions before 9:00pm, send a private message to the host of your Zoom call (“Ready to scan”) and wait for confirmation before leaving the Zoom call.
- If something goes seriously wrong during the exam, send email to Timothy (tmc@illinois.edu) and the proctor as soon as possible explaining the situation. If you have already finished the exam but cannot submit to Gradescope for some reason, include a complete scan of your exam as a PDF file in your email. If you are in the middle of the exam, send email to Timothy and the proctor, continue working until the time limit, and then send a second email with your completed exam as a PDF file. Please do not email raw photos.
- You are reminded about the course’s, the department’s, and the university’s academic integrity policies.
- Avoid the deadly sins. Write complete solutions, not examples. Declare all your variables. Write concisely and precisely.
- Good luck!

1. [28 PTS] *Short questions.*

(a) [6 PTS] Give an asymptotically tight solution to the following recurrence:

$$T(n) = \begin{cases} 3T(\lfloor n/2 \rfloor) + n^{3/2} & \text{if } n \geq 2022 \\ 374 & \text{if } n < 2022. \end{cases}$$

Justify your answer.

(b) [3 PTS] True or False: quicksort is always faster than mergesort. Briefly justify your answer.

(c) [3 PTS] Consider the following pseudocode to compute the Fibonacci number F_n :

1. $X = 0, Y = 1$
2. for $i = 2$ to n do
3. $Z = X, X = Y, Y = Z + Y$
4. return Y

What is the running time as a function of n , assuming that arithmetic operations take constant time?

(d) [3 PTS] In the same pseudocode as (c), what is the running time as a function of n , when we take into account the bit complexity of the numbers? Justify your answer.

(e) [3 PTS] Given an undirected graph $G = (V, E)$, consider a BFS tree T . True or False: There cannot be an edge between a vertex at level i to a vertex with level at $i + 3$ of the tree T . If true, justify your answer; if false, give a counterexample.

(f) [3 PTS] For a DAG G with n vertices, what is the minimum number of strongly connected components that G can have as a function of n ? Explain.

(g) [3 PTS] True or False: In an undirected graph G where every vertex has degree at least two, G must have a cycle. Justify your answer.

(h) [4 PTS] Which algorithm from class gives the most efficient solution to the all-pairs shortest paths (APSP) problem in the case when the input graph has $m = n^{3/2}$ edges, and all edges have positive weights? (Here, n denotes the number of vertices.) Justify your answer.

2. [20 PTS] We are given a list of n exam scores $\langle a_1, \dots, a_n \rangle$; the list is **not** sorted. We are also given a value S . We want to find the smallest k such that the sum of the top k scores is at least S .

For example, for $\langle 51, 93, 62, 100, 31, 77, 85, 49 \rangle$ and $S = 275$, then the answer is $k = 3$, since the top 3 scores 100, 93, 85 sum to 278, which is greater than S .

Describe an algorithm to solve this problem in strictly better than $O(n \log n)$ time. Give pseudocode using recursion, and analyze its running time using a recurrence. You may use the linear-time median finding algorithm from class as a subroutine.

3. [28 PTS] Consider the following problem:

Given a sequence of points $\langle p_1, \dots, p_n \rangle$ in two dimensions and a value L , find a subsequence $\langle p_{j_1}, \dots, p_{j_m} \rangle$ (with $j_1 < j_2 < \dots < j_m$) that maximizes the length m subject to the constraint that $d(p_{j_i}, p_{j_{i+1}}) \leq L$ for all $i \in \{1, \dots, m-1\}$. (Here, $d(p, q)$ denotes the Euclidean distance between p and q , which is computable in $O(1)$ time.)

Consider the following dynamic programming approach. Define the following subproblems: for each $i \in \{0, \dots, m\}$ and $j \in \{0, \dots, n\}$, let $C(i)$ be the length of the longest subsequence of $\langle p_i, \dots, p_n \rangle$ that starts at p_i and satisfies the above distance constraint.

We have the following recursive formula: for each $i \in \{1, \dots, m\}$,

$$C(i) = \max \left\{ 1, \max_{j \in \{i+1, \dots, n\} \text{ such that } d(p_i, p_j) \leq L} (C(j) + 1) \right\}.$$

(You may assume correctness of the above formula without proof.)

- (a) [15 PTS] Write pseudocode to solve the problem using the above formula. Your pseudocode should output the length of the longest subsequence of the original problem (you are not required to output the optimal subsequence itself). Analyze the running time of your pseudocode as a function of n (it must run in polynomial time).

- (b) [13 PTS] Consider the following modified problem:

Given a sequence of points $\langle p_1, \dots, p_n \rangle$ in two dimensions, a value L , and an integer r , find a subsequence $\langle p_{j_1}, \dots, p_{j_m} \rangle$ (with $j_1 < j_2 < \dots < j_m$) that maximizes $\sum_{i=1}^{m-1} d(p_{j_i}, p_{j_{i+1}})$, such that $|\{i \in \{1, \dots, m-1\} : d(p_{j_i}, p_{j_{i+1}}) > L\}| = r$.

Describe a dynamic programming algorithm to solve this modified problem, i.e., give a new definition of subproblems, give recursive formulas (including base cases, and how to obtain the optimal value), and specify the evaluation order. You do not need to give pseudocode for this part of the question (and you do not need to give justifications of your formula if it is clear enough). Analyze the running time of your algorithm as a function of n and L .

[Hint: you may want to add one more parameter to $C(\cdot)$, i.e., use a two-dimensional table.]

4. [24 PTS]

- (a) [9 PTS] We are given a DAG G with n vertices and m edges, where each edge is colored red or blue. We are also given two vertices s and t .

We want to decide whether there is a path from s to t such that at least 25% of the edges in the walk are red.

Describe an efficient solution to this problem by applying a known algorithm. Bound the running time.

[Hint: if it is helpful, perhaps warm up with a variant of the problem with 25% changed to 50%.]

- (b) [15 PTS] We are given a directed graph G with n vertices and m edges. We are also given two vertices s and t , where each *vertex* (not edge) is colored red or blue.

We want to compute a walk from s to t that has strictly more red vertices than blue vertices, while minimizing the total number of red vertices.

Describe an efficient solution to this problem by constructing a new graph and applying a known algorithm. Remember to clearly define the vertices and edges of your new graph, and analyze the total running time.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs