CS/ECE 374 A ✧ Fall 2026

Midterm 2 Practice 3

November 8, 2026

Name:	Aashima Sisodia
NetID:	aashima2

- ***Don't panic!***
- You have 120 minutes to answer five questions. **The questions are described in more detail in a separate handout.**
- If you brought anything except your writing implements, your **hand-written** double-sided 8½" × 11" cheat sheet, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- Please clearly print your name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- Greedy algorithms require formal proofs of correctness to receive credit. Otherwise, proofs or other justifications beyond those listed in the standard grading rubrics are required if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.

- **Do not write outside the black boxes on each page**. These indicate the area of the page that our scanners will actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, feel free to use the overflow/scratch pages at the back of the answer booklet, but **clearly indicate where we should look**. If we can't find your work, we can't grade it.
- **Only work that is written into the stapled answer booklet will be graded**. We **do not** recommend detaching pages from the answer booklet, because any work on those detached pages will not be graded (and it disrupts our process for uploading solutions to Gradescope.) Please let us know if you detach a page either on purpose or accidentally so we can reattach it.
- We will provide additional blank scratch paper on request, but we will not grade anything written on it.
- **Please return all paper to Emily or a TA** including your **answer booklet**, your question sheet, your cheat sheet, and all scratch paper. There will be two piles of collected paper, the first for answer booklets and the second for everything else. We **will not** consider the paper in the second pile when grading your exam.

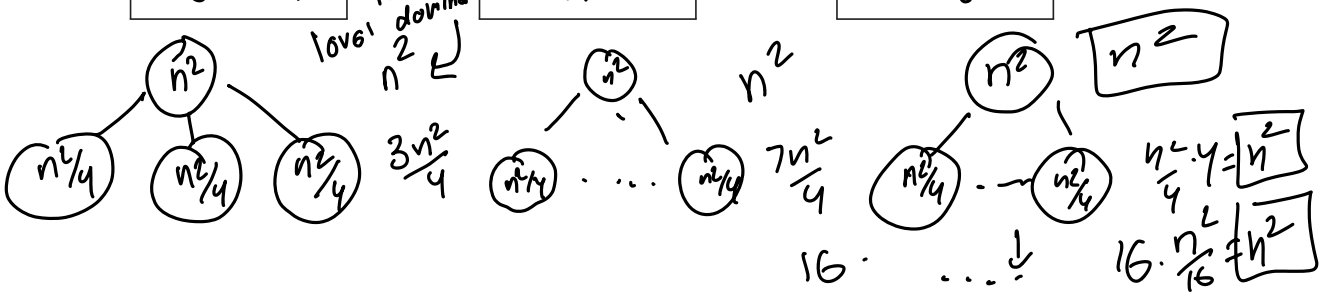
This sentence contains four and three fourths percent a's, five and one half percent c's, three percent d's, eighteen and one fourth percent e's, four and one half percent f's, three fourths percent g's, five percent h's, two and one half percent i's, two percent l's, twelve and one half percent n's, six percent o's, five percent p's, eight percent r's, seven percent s's, nine and three fourths percent t's, two and one fourth percent u's, one and one half percent v's, one and one fourth percent w's and one half percent x's.

CS/ECE 374 A ✧ Fall 2026 Midterm 2 Practice 3 Problem 1	Name: Aashima Sisodia

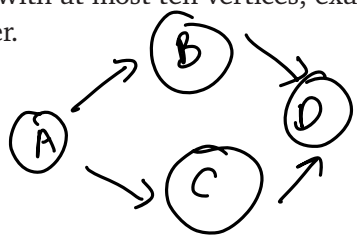
(a) Write the solution to each of the following recurrences in the box immediately below it. (Use the space below the boxes for scratch work.)

$A(n) = 3A(n/2) + O(n^2)$ $B(n) = 7B(n/2) + O(n^2)$ $C(n) = 4C(n/2) + O(n^2)$

$O(n^2)$ | $n^{\log_2 7}$ | $n^2 \log n$

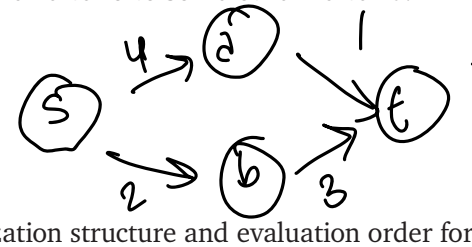


(b) Draw a directed acyclic graph with at most ten vertices, exactly one source, exactly one sink, and more than one topological order.



s, a, b, t and s, b, a, t are both valid topo order

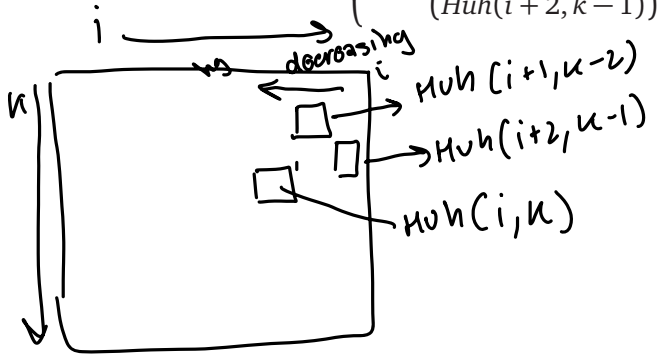
(c) Draw a directed graph with at most ten vertices, with distinct positive edge weights, that has more than one shortest path from some vertex s to some other vertex t.



both have length 5

(d) Describe an appropriate memoization structure and evaluation order for the following (meaningless) recurrence, and give the running time of the resulting iterative algorithm to compute $Huh(1, n)$.

$$Huh(i,k) = \begin{cases} 0 & \text{if } i > n \text{ or } k < 0 \\ \min \begin{Bmatrix} Huh(i+1,k-2) \\ Huh(i+2,k-1) \end{Bmatrix} + A[i,k] & \text{if } A[i,k] \text{ is even} \\ \max \begin{Bmatrix} Huh(i+1,k-2) \\ Huh(i+2,k-1) \end{Bmatrix} - A[i,k] & \text{if } A[i,k] \text{ is odd} \end{cases}$$




```
for i = n+1 to 1:
    for k = 0 to n:
```

$O(n^2)$

CS/ECE 374 A ✧ Fall 2026 Midterm 2 Practice 3 Problem 2	Name: Aashima 2

See the question sheet for a detailed description of your game with Elmo.

(a) ***Prove*** that you should not also use the greedy strategy. That is, show that there is a game that you can win, but only if you do *not* follow the same greedy strategy as Elmo. Assume Elmo plays first.

(b) Describe and analyze an algorithm to determine, given the initial sequence of cards, the maximum number of points that you can collect playing against Elmo.

(a) Take the cards [6, 5, 100, 3, 4] with Elmo moving first. Elmo takes 6, leaving 5, 100, 3, 4.

If we play greedy: I take 5, Elmo takes 100, I take 4, and Elmo plays 3. I score 9, and Elmo scores 109. We lose

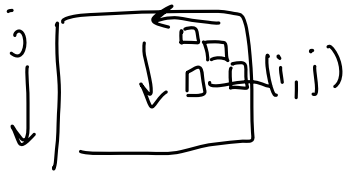
However, if we do not follow this strategy and play smartly, I can take 4, Elmo takes 5. I take 100, and Elmo takes 3. I score 104, and Elmo 104.

(b) Let Best(i, j) be the most points one can collect from cards A[i...j] when it is Elmo's turn.

$$Best(i,j) = \begin{cases} 0 & \text{if } i \geq j \\ \max \begin{bmatrix} A[i+1] + Best(i+2, j) \\ A[j] + Best(i+1, j-1) \end{bmatrix} & \text{if } A[i] > A[j] \\ \max \begin{bmatrix} A[i] + Best(i+1, j-1) \\ A[j-1] + Best(i, j-2) \end{bmatrix} & \text{otherwise} \end{cases}$$

The top-level call is Best(1, n)

Eval order:




```
for i=n to 1:
    for j=1 to n:
```

$O(n^2)$ loop

$O(1)$ subproblem work

Time complexity: $O(n^2)$

CS/ECE 374 A ✧ Fall 2026 Midterm 2 Practice 3 Problem 3	Name:

Suppose you are given a directed graph $G = (V, E)$, whose vertices are either red, green, or blue. Edges in G do not have weights, and G is not necessarily a dag. The ***remoteness*** of a vertex v is the *maximum* of three shortest-path lengths:

- The length of a shortest path to v from the closest red vertex
- The length of a shortest path to v from the closest blue vertex
- The length of a shortest path to v from the closest green vertex

In particular, if v is not reachable from vertices of all three colors, then v is infinitely remote. Describe and analyze an algorithm to find a vertex of G with *minimum* remoteness.

Defino directed grap $G' = (V', E')$:

- $V' = V \times \{R, G, B\} \cup \{z\}$. We construct a copy of every vertex per color. z is a new source node.

$|V'| = 3|V| + 1$

- $E' = \{(u, c) \to (v, c) \mid (u \to v) \in E \text{ and } c \in \{R, G, B\}\}$ representing each layer

$\cup$

$\{z \to (v, R) \mid v \in V \text{ is red}\}$

$\cup$

$\{z \to (v, G) \mid v \in V \text{ is green}\}$

$\cup$

$\{z \to (v, B) \mid v \in V \text{ is blue}\}$

$|E'| = 3|E| + |V|$

As a result of this construction, any path from z first enters exactly one layer C, at a vertex of color C. The rest of the path is within that color layer. Since we need to compute the shortest path in this unweighted graph, we call BFS(z). For each $v \in V$, we compute remoteness(v) = max over $c \in \{R, G, B\}$ of dist$(z, (v, c))$ − 1. Return vertex with smallest remoteness

(continue on page 6)

CS/ECE 374 A ✧ Fall 2026 Midterm 2 Practice 3 Problem 4	Name: Aashima Sisodia

Suppose you are given an array $A[1..n]$ of integers such that $A[i]+A[i+1]$ is even for *exactly one* index i. In other words, the elements of A alternate between even and odd, except for exactly one adjacent pair that are either both even or both odd. Describe and analyze an efficient algorithm to find the unique index i such that $A[i]+A[i+1]$ is even.

```
FindPair(lo, hi):
    if hi = lo + 1:
        return lo
    mid ← ⌊(lo + hi)/2⌋
    if A[mid] + A[1] + mid - 1 is even:
        return FindPair(mid, hi)
    else:
        return FindPair(lo, mid)
```

(or: if hi - lo < 374: brute force)

FindPair(1, n) returns the unique index i such that A[i] + A[i+1] is even. Since the prefix 1...i should still be alternating, we need to find where the pattern breaks in parity.

We shrink our search space in the following manner:

- if A[mid] + A[1] + mid - 1 is even, then it conforms to the alternating pattern. So mid < i and we are still in the prefix
- if instead it is odd we are in the suffix. So mid > i & so we must search in the earlier half.

Running time:

modification of binary search

$(T(n) = T(n/2) + O(1))$

so $O(\log n)$

CS/ECE 374 A ✧ Fall 2026 Midterm 2 Practice 3 Problem 5	Name:

A *zigzag walk* in in a directed graph G is a sequence of vertices connected by edges in G, but the edges alternately point forward and backward along the sequence. Specifically, the first edge points forward, the second edge points backward, and so on. The *length* of a zigzag walk is the sum of the weights of its edges, both forward and backward.

Suppose you are given a directed graph G with non-negatively weighted edges, along with two vertices s and t. Describe and analyze an algorithm to find the shortest zigzag walk from s to t in G.

Define a directed graph $G' = (V', E')$:

- Vertices: $V' = V \times \{F, B\}$. (v, F) means that the next edge must be forward. (v, B) means that the next edge must be traversed backwards.
- Edges: For $u \to v \in G$ with weight w,

$E' = \{(u,F) \to (v,B) \mid u \to v \in E\}$ (with weight w) indicating a forwards walk $\cup$ $\{(u,B) \to (v,F) \mid u \to v \in E\}$ (with weight w) indicating a reversed edge

Note that $|V'| = 2|V|$ and $|E'| = 2|E|$. Every walk in G' from (s,F) alternates between the F layer and B layer. So its edges correspond to edges of G that alternatively point forwards and backwards, starting forwards.

Since edge weights are non-negative, we run Dijkstra's algorithm on G' from (s,F) and return the smaller of dist$((s,F),(t,F))$ and dist$((s,F),(t,B))$ to represent the shortest zigzag walk.

Runtime: Dijkstra's is $O(|E'| \log |V'|) = O(2|E| \log 2|V|) = O(E \log V)$.

(scratch paper)

Running time: Since $|V'| = 3|V|+1$ and $|E'| = 3|E|+|V|$ BFS takes $O(|V'|+|E'|) = O(3|V|+1+3|E|+|V|) = O(|V|+|E|)$.

(scratch paper)

(scratch paper)

(scratch paper)

(scratch paper)