

Write your answers in the separate answer booklet.

You have 120 minutes (after you get the answer booklet) to answer five questions.

Please return this question sheet and your cheat sheet with your answers.

1. Short answers:

(a) Solve the following recurrences:

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

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

(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 $Huh(1, n)$.

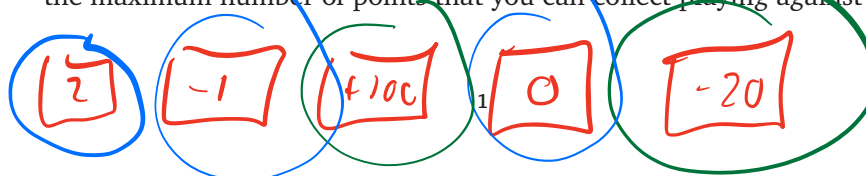
$$Huh(i, k) = \begin{cases} 0 & \text{if } i > n \text{ or } k < 0 \\ \min \left\{ \begin{array}{l} Huh(i+1, k-2) \\ Huh(i+2, k-1) \end{array} \right\} + A[i, k] & \text{if } A[i, k] \text{ is even} \\ \max \left\{ \begin{array}{l} Huh(i+1, k-2) \\ Huh(i+2, k-1) \end{array} \right\} - A[i, k] & \text{if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

2. You and your eight-year-old nephew Elmo decide to play a simple card game. At the beginning of the game, the cards are dealt face up in a long row. Each card is worth a different number of points, which could be positive, negative, or zero. After all the cards are dealt, you and Elmo take turns removing either the leftmost or rightmost card from the row, until all the cards are gone. At each turn, you can decide which of the two cards to take. The winner of the game is the player that has collected the most points when the game ends.

Having never taken an algorithms class, Elmo follows the obvious greedy strategy—when it's his turn, Elmo *always* takes the card with the higher point value. Your task is to find a strategy that will beat Elmo whenever possible. (It might seem mean to beat up on a little kid like this, but Elmo absolutely *hates* it when grown-ups let him win.)

(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.



3. 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:

max min

- 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.

goal: min max min

4. 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. For example, given the following array as input, your algorithm should return the integer 6, because $A[6] + A[7] = 88 + 62$ is even. (Cells containing even integers are shaded blue.)

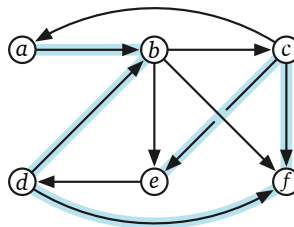
17	40	23	72	39	88	62	13	40	53	92	21	10	73	68
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Try for $O(n)$ if we can,

5. A zigzag walk 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.

For example, the following graph contains the zigzag walk $a \rightarrow b \leftarrow d \rightarrow f \leftarrow c \rightarrow e$. Assuming every edge in the graph has weight 1, this zigzag walk has length 5.



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 .

Two layers $(s, t) + (s, b)$

Dijkstra $((s, t))$,

for $(t, f) + (t, b)$

$\text{FindI}(A[1..n]) \leftarrow$ Returns i as defined
by problem

CS/ECE 374 A ✧ Fall 2025
☞ Midterm 2 Practice 1 ☞
November 6, 2025

Name:	Emily Fox
NetID:	ekfox

-
- **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 $8\frac{1}{2}'' \times 11''$ 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 any credit, even if they are correct. ~~Otherwise, proofs or other justifications are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.~~
-
- **Please do not write outside the black boxes on each page.** These indicate the area of the page that our scanners can actually scan. If the scanner can't see your work, we can't grade it.
 - If you run out of space for an answer, please use the overflow/scratch pages at the back of the answer booklet, but **please 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.** In particular, you are welcome to detach scratch pages from the answer booklet, but any work on those detached pages will not be graded. We will provide additional scratch paper on request, but any work on that scratch paper will not be graded.
-

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.

- (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.)

$$A(n) = 3A(n/2) + O(n^2)$$

$$O(n^2)$$

$$B(n) = 7B(n/2) + O(n^2)$$

$$O(n^{\log_2 7})$$

$$C(n) = 4C(n/2) + O(n^2)$$

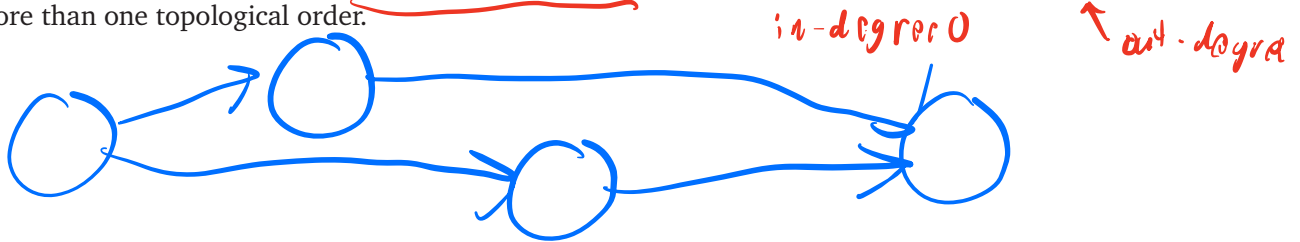
$$O(n^2 \log n)$$

$$\begin{array}{c} n^2 \\ \swarrow \quad \downarrow \quad \searrow \\ n^2/4 \quad n^2/4 \quad n^2/4 \end{array} = \frac{3}{4} n^2$$

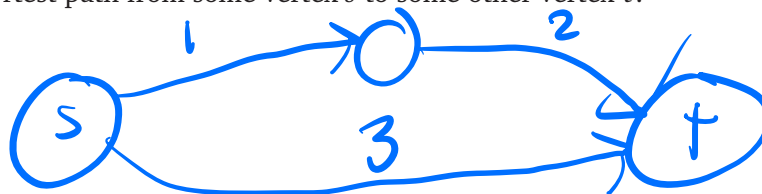
$$\begin{array}{c} n^2 \\ \swarrow \quad \downarrow \quad \searrow \\ n^2/4 \quad n^2/4 \quad n^2/4 \end{array} \times 7^{\log_2 n} = \frac{7}{4} n^2$$

$$\begin{array}{c} n^2 \\ \swarrow \quad \downarrow \quad \searrow \\ n^2/4 \quad n^2/4 \quad n^2/4 \end{array} \times 4 = n^2$$

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



- (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 $Huh(1, n)$.

$$1 \leq i \leq n+2$$

$$-2 \leq k \leq n$$

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

Structure: 2D array $Huh[1..n+2, -2..n]$.

for $i \in n+2$ to 1

$k \leftarrow -2$ to n

$O(1)$ per

$O(n^2)$ problems:

$O(n^2)$ time



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) $\boxed{2} \quad \boxed{-1} \quad \boxed{100} \quad \boxed{0} \quad \boxed{-20}$ If I'm greedy
 $\downarrow \quad \times \quad \downarrow \quad \times \quad \downarrow$ Elmo: 2, 100, -20
 Me: -1, 0

(b) Assume cards are $A[1..n]$ in left-to-right order on my first turn.
 Better: Elmo 2, 0, -1
 Me: -20, 100

$MP(i, j)$: Max points I can get starting with my turn & cards $A[i..j]$ remaining.

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

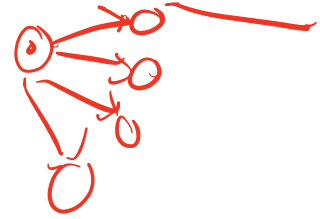
Time: $O(n^2)$

Cont. on page 6.

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

do all v at once?



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.

Let $G' = (V', E')$. $V' = V \cup \{s_r, s_g, s_b\}$

$E' = E \cup \{s_r \rightarrow v \mid v \text{ is red}\}$

Remoteness of v is $\max\{ \text{dist}(s_r, v), \text{dist}(s_g, v), \text{dist}(s_b, v) \}$ - 1, $\cup \{s_g \rightarrow v \mid v \text{ is green}\}$
 $\cup \{s_b \rightarrow v \mid v \text{ is blue}\}$

Alg: Run BFS(s_r), BFS(s_g), BFS(s_b).

Use distances to compute remoteness
for all vertices in $O(1)$ time each.

Return vertex with min remoteness.

Time: $O(V' + E') = O(V + E) + V$

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.

Answer is in $A[low..hi]$

2 3 7 10

if $hi - low \neq A[hi] + A[low] \mod 2$

$low \leftarrow 1$

$hi \leftarrow n$

while $low \leq hi - 2$

$mid \leftarrow \left\lfloor \frac{low + hi}{2} \right\rfloor$

if $hi - mid \neq A[hi] - A[mid] \mod 2$

$low \leftarrow mid$

else

$hi \leftarrow mid$

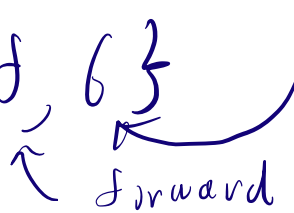
Time: Binary search
so $O(\log n)$

return low

A zigzag walk 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 .

Let $G' = (V', E')$ s.t.

$V' = V \times \{f, b\}$  backward next
forward next

$E' = \{(u, f) \rightarrow (v, b) \mid u \rightarrow v \in E\}$

$\cup \{(v, b) \rightarrow (u, f) \mid u \rightarrow v \in E\}$

Want shortest path from (s, f) to
either of (t, f) or (t, b) .

Run Dijkstra $((s, f))$ + return shortest
path to smaller distance of
 (t, f) or (t, b) with second component
stripped out. Time: $O(E' \log V')$

$= O(E \log V)$

(scratch paper)

★
2 (6) continued... $1 \leq i \leq n+1$
 $0 \leq j \leq n$

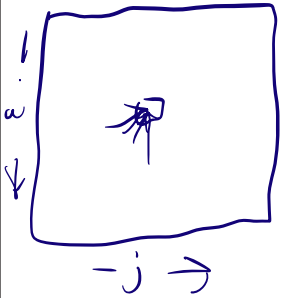
Memoize with 2D array $MP[1..n+1, 0..n]$.

Eval order: for $i \leftarrow n+1$ down to 1

for $j \leftarrow 0$ to n

$O(1)$ per subproblem \times ($O(n^2)$ subproblems)

$= O(n^2)$ time



(scratch paper)

(scratch paper)

(scratch paper)

(scratch paper)