

WRIGHT STATE UNIVERSITY
Department of Computer Science and Engineering
CS7200: Algorithm Design and Analysis

Spring 2020

FINAL (50 pts)

Prasad

Be concise and precise.

NAME	
UID	
EMAIL	
POINTS	

1) [3 + 3 + 3 + 3 + 3 = 15pts] [Solving Recurrences: Big-Oh]

Provide “best” Big-Oh bound from among $\{O(n \log n), O(n \log^2 n), O(n), O(n^2), O(n^3), O(n^4), O(n^n), \text{NONE}\}$ for each of the following recurrences. Assume $T(1) = 1$. Choose *NONE* only if the other alternatives in the set cannot upper bound the growth of the solution to the recurrence.

$$(a) \quad T(n) = 2 T(n/4) + n^2$$

$$\mathbf{T(n) =}$$

$$(b) \quad T(n) = 2 T(n/2) + n \log n$$

$$\mathbf{T(n) =}$$

$$(c) \quad T(n) = 9 T(n/3) + n$$

$$\mathbf{T(n) =}$$

$$(d) \quad T(n) = n * T(n-1)$$

$$\mathbf{T(n) =}$$

$$(e) \quad T(n) = T(n-1) + n^3$$

$$\mathbf{T(n) =}$$

2) [5 + 2 + 6 + 2 = 15 pts][Dynamic Programming: String Matching]

Use dynamic programming technique to find least-cost alignment of the two strings MARS and CAAS under the operation cost model where *insert* and *delete* cost +1, *substitute* costs +2 and *match* costs -2. Clearly state (i) the recurrence relation and then (ii) fill the appropriate table given on the *next page*, and show the least-cost alignment (cost plus properly aligned strings) *on this page* below? (There is a third part of the question *on the next page*.)

(i) RECURRENCE RELATION WITH BOUNDARY CONDITIONS AND COST MODEL INCORPORATED:

(ii) RESULT → MINIMUM COST AND ALIGNMENT:

STRINGS					

USE THE ABOVE TABLE AS TEMPORARY SCRIBBLE SPACE.

(ii)

WRITE THE FINAL TABLE BELOW (COPY COST AND ALIGNMENT TO THE PREVIOUS PAGE):

STRINGS	e	M	A	R	S
e					
C					
A					
A					
S					

(iii) Is the least-cost alignment unique? CIRCLE ONE OF: **YES** **NO**

3) [10 pts] [Divide and Conquer: Reverse Engineering Moves]

Let $IA[1 \dots n]$ be an array of distinct sorted integers that has been moved around m locations to the **left**. For example, $[67, 279, 356, 400, 420, -55, -5]$ is a sorted array that has been moved around $m = 2$ locations to the left, while $[356, 400, 420, -55, -5, 67, 279]$ has been moved around $m = 4$ locations to the left.

Design an efficient algorithm that takes the array IA and determines m . Specifically, provide pseudocode (optionally, with annotations/comments) that makes your intent clear. (Make sure to informally verify it on a small example to improve your faith in your code.) You may assume n to be a power of two.

[Hint:

- (i) Define a procedure that takes three arguments: an array IA , and two indices *first* and *last* into the array. (Alternatively, you may define IA to be global and reduce one argument.)
- (ii) In the body, compare the first element and the middle element of the array and then the middle two elements to generate subproblems.]

/* Provide a brief description here and show your pseudocode on the next page. Do not let the pseudocode span two pages. */

4) [2 * 5 = 10pts][Intractability]

Highlight/Boldface/Color Green/Retain correct responses.

- (a) Assume you have been given an encoding of an instance of VERTEX-COVER problem as an instance of INDEPENDENT-SET problem. What is the “planned” direction of the polynomial time reduction?

VERTEX-COVER $\leq_p$ INDEPENDENT-SET or
INDEPENDENT-SET $\leq_p$ VERTEX-COVER

- (b) If VERTEX-COVER is intractable, can we conclude that INDEPENDENT-SET is intractable?

YES or NO

- (c) If INDEPENDENT-SET is intractable, can we conclude that VERTEX-COVER is intractable?

YES or NO

- (d) If VERTEX-COVER is efficiently solvable, can we conclude that INDEPENDENT-SET is efficiently solvable?

YES or NO

- (e) If INDEPENDENT-SET is efficiently solvable, can we conclude that VERTEX-COVER is efficiently solvable?

YES or NO