

## FINAL EXAM REVIEW

**Problem 1: Asymptotics**

Consider the set  $S = \{a, b, c, d, e\}$  of five functions  $a$  through  $e$ , which are classified as follows:

- $a$      $(1)$
- $b$      $(\lg n)$
- $c$      $(n)$
- $d$      $(n \lg n)$
- $e$      $(n^2)$

In the following table, each row is labelled by an asymptotic notation that designates a set of functions  $R$ . For each row, indicate the result of the intersection  $R \cap S$ , i.e., the set of all functions in  $S$  that are also members of  $R$ . (Each entry is worth 2 points.)

<b>R</b>	<b><math>R \cap S</math></b>
$(n \lg n)$	
$O(n)$	
$o(\lg n)$	
$(1)$	
$(n^2)$	

## Problem 2: Asymptotics

For each entry (*row*, *col*) in the following table, determine whether membership in the set labelled *row* implies membership in the column labelled *col*. Fill in each entry of the table with one of the following three answers:

- *Yes*: A member of set *row* is definitely a member of set *col*.
- *No*: A member of set *row* is definitely not a member of set *col*.
- *Maybe*: Some, but not all, members of set *row* are members of set *col*.

	$(n \lg n)$	$(n \lg n)$	$(n \lg n)$	$(n \lg n)$	$(n \lg n)$
$o(n)$					
$o(n^2)$					
$O(n)$					
$O(n^2)$					
$(n)$					
$(n^2)$					
$(n)$					
$(n^2)$					
$(n)$					
$(n^2)$					

## Problem 3: Recurrences

Use  $\Theta$  notation to describe solutions to each of the following recurrence equations. You may assume in all cases that  $T(n) = 1$  for  $n = 1$ .

- $T(n) = T(n - 3) + 2$
- $T(n) = T(n - 7) + 3n$
- $T(n) = T(n/3) + 5$
- $T(n) = T(n/5) + 2n$
- $T(n) = 2T(n/2) + 5$
- $T(n) = 2T(n/2) + 3n$

### Problem 4: Recurrences, Dynamic Programming

Given a numeric array  $A$ , the *partial sum array*  $PSA(A)$  is an array  $P$  that has the same length as  $A$  in which  $P[i]$  is the sum of all the elements in the array segment  $A[1..i]$ . For example, the following figure show an array  $A$  and its partial sum array  $P = PSA(A)$ :

A	2	5	3	6	1	4
P	2	7	10	16	17	21

Here is one algorithm that correctly computes the partial sum array  $P$  for an array  $A$ :

```
PSA1(A)
  P <- MakeArray(length[A]) {Make a new array for the result.}
  for i <- 1 to length[A] do
    P[i] <- Segment-Sum(A, 1, i)
  return P

Segment-Sum(A, lo, hi)
  if lo > hi
    then return 0
    else return A[hi] + Segment-Sum(A, lo, hi - 1)
```

**Part a.** Write a recurrence equation  $T(n)$  that expresses the *worst-case* running time of  $PSA1$  on an array of length  $n$ .

**Part b.** Solve the recurrence equation from part a to obtain a asymptotic worst-case time bound using  $\Theta$ .

**Part c.** For every  $i, 1 \leq i \leq \text{length}[A]$ ,  $PSA(A)$  calculates  $\text{Segment-Sum}(A, 1, i)$   $i$  times. This is wasteful. Using dynamic programming techniques, develop a  $\Theta(n)$  algorithm  $PSA2(A)$  for computing the partial sum array of  $A$ . Write the pseudocode for  $PSA2$  and show that it runs in worst-case time  $\Theta(n)$ . Hint: Use the resulting array  $P$  as the “table”.

### Problem 5: Probability

Suppose you are given a black-box procedure  $\text{Roll6}()$  that simulates rolling a fair die. That is,  $\text{Roll6}()$  returns an integer in the range  $[1..6]$ , with each integer being equally likely.

- Using  $\text{Roll}()$ , write pseudocode for a function  $\text{Roll7}()$  that returns an integer in the range  $[1..7]$ , with each integer being equally likely.
- What is the expected number of times that  $\text{Roll6}()$  is called for a single top-level call of  $\text{Roll7}()$ .

### Problem 6: Sorting

a. Assume you want to sort an array of  $n$  integers taken from the range  $[1..n]$ . For each of the following algorithms, indicate the asymptotic (1) worst-case running time (2) best case running time and (3) average-case running time:

- selection sort
- insertion sort
- bubble sort
- merge sort
- quick sort
- heap sort
- tree sort (insert elements into binary search tree, and extract them via in-order traversal.)
- counting sort
- radix sort
- bucket sort

b. Which of the above methods are not viable if the array may contain any numbers in the range  $[1..n]$ , not just integers?

### Problem 7: Trees

Suppose that a tree  $T$  is both a heap and a binary search tree. At most how many nodes can  $T$  have?

### Problem 8: Trees

For each of the following types of trees, indicate the minimum possible height and maximum possible height as a function of  $n$ , the number of nodes in the tree. (Do *not* use asymptotic notation!)

- a. binary tree
- b. binary search tree
- c. red-black tree
- d. heap

### Problem 9: Greediness (CLR 17.2-3)

Suppose that in a 0-1 knapsack problem (i.e. you either take an entire item or not), the order of the items when sorted by increasing weight is the same as their order when sorted by decreasing value. Give an efficient algorithm to find an optimal solution to this variant of the knapsack problem, and argue that your algorithm is correct.

### Problem 10: Graphs

- a. Draw a single directed graph  $G$  that has all of the following features:
- 5 vertices =  $\{A, B, C, D, E\}$
  - 8 edges
  - 3 strongly connected components
  - 2 weakly connected components
- b. Can you perform a topological sort on your graph  $G$  from part a? Explain.
- c. Show the tree induced by depth-first search of  $G$ . Assume that vertices are explored in alphabetical order and that adjacency lists are stored in alphabetical order.
- d. Show the tree that is induced by breadth-first search of  $G$ .
- e. Assign weights to the edges of  $G$ , and show a tree induced by Dijkstra's single-source shortest path algorithm starting at a vertex in the largest weakly connected component.
- f. Consider the weighted undirected graph  $G'$  derived from the weighted directed graph  $G$  of part e by erasing the directions on all edges. Show a tree induced by Prim's minimum spanning tree algorithm.

### Problem 11: Complexity

Consider the following language:

$$K\text{-CYCLE} = \{ \langle G, k \rangle \mid G \text{ is an encoding of a directed graph} \\ \text{that contains a simple cycle with } k \text{ vertices.} \}$$

(Recall that a cycle is a path = a sequence of vertices  $v_0, v_1, \dots, v_k$  in which the  $v_0 = v_k$ . A simple cycle is one in which  $\{v_1, \dots, v_k\}$  are distinct.)

- a. In English, describe the elements of  $\overline{K\text{-CYCLE}}$ .
- b. Show that  $K\text{-CYCLE}$  is in NP.
- c. Show that  $K\text{-CYCLE}$  is NP-complete by finding an appropriate reduction between  $\text{HAM-CYCLE}$  and  $K\text{-CYCLE}$ .
- d. Explain why most computer scientists consider it unlikely that membership in  $K\text{-CYCLE}$  can be determined in polynomial time.
- e. Suppose that you discover a polynomial-time algorithm for determining if  $\langle G, k \rangle$  is a member of  $K\text{-CYCLE}$ . Explain why this would imply that  $P = NP$ .
- f. What would you have to do to prove that  $P = NP$ ?

### **Problem 12: Compression**

Consider the phrase TO BE OR NOT TO BE.

- a. How many bits does it take to encode the above phrase using a fixed length character encoding? Assume that space characters need to be encoded. Do *not* include the bits needed to represent the table mapping each character to its fixed length encoding.
- b. How many bits does it take to encode the above phrase using the variable length character encoding produced by a Huffman code? Do *not* include the bits needed to represent the table mapping each character to its variable length encoding. Show your work.

### **Problem 13: Compression**

Ann has a small Pascal program PROG that generates a huge output file OUT when applied to a small input file IN. Bob, who lives far away from Ann, needs a copy of OUT so that he can analyze it as part of his research. Briefly describe a very simple compression scheme that allows Ann to send Bob a highly compressed version of OUT.