Version 3

CSE Ph.D. Qualifying Exam, Spring 2020
Theory & Algorithms

PRINT NAME: ______________________________

PRINT PSU EMAIL: ______________________________ PSU ID: ____________

SIGNATURE: ______________________________

INSTRUCTIONS: **Do 5 of 6 problems.** ***Indicate on the grading table below which problem is NOT to be graded by crossing out the corresponding column in the grade box below*, otherwise an arbitrary question will be chosen not to grade.** No aids allowed. Neatly and coherently **justify** your answers in the space allotted. Use the back of a page or facing page if necessary and, if you do, clearly mark "continue on back" or "continue on facing page."

QUESTION:	1	2	3	4	5	6	TOTAL
SCORE:							
MAXIMUM:	20	20	20	20	20	20	100

1. Recursions [$5 + 2 + 3 + 10 = 20$ points]

   (a) Suppose a staircase can be climbed by a combination of single steps (taken one at a time) and double steps (taking steps two at a time). Let c_n be the number of different ways (where order matters!) to climb a staircase of $n \geq 1$ stairs.

      i. Find *and explain* a recurrence relation for the c_n, $n \geq 1$.

      ii. Specify the numerical values of c_n for all $n \leq 5$.

(b) Consider an ordered row of n buckets and an unlimited number of identical red balls, where each bucket is just large enough to contain only one ball. Let s_n be the number of ways that balls can be assigned to buckets so that balls are never assigned to adjacent buckets (*not* including the case where none of the buckets are assigned a ball).

i. Specify the numerical values of s_n for all $n \leq 5$.

ii. Find *and explain* a recurrence relation for the s_n, $n \geq 1$.

2. Induction [$3 + 2 + 15 = 20$ points]

(a) Consider a predicate $P(\cdot)$. Suppose that we have proved $P(1)$ and $P(3)$. Find the set of integers k for which $P(k)$ is proved true if we also prove that: $\forall k \geq 1,\ P(k+1) \Rightarrow P(k) \wedge P(k+3)$.

(b) Suppose instead that we have proved $P(-2)$ and $P(0)$. Find the set of integers k for which $P(k)$ is proved true if we also prove that: $\forall k \geq 1,\ P(k) \wedge P(k+2) \Rightarrow P(k-2)$.

(c) Prove by induction that, $\forall n \geq 1$

$$1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \frac{1}{\sqrt{4}} + \ldots + \frac{1}{\sqrt{n}} \ > \ 2(\sqrt{n+1} - 1).$$

Hint: $\sqrt{2} \approx 1.4$.

3. Quantifiers [$10 + 10 = 20$ points]

(a) The first three of the following statements are premises and the fourth is a conclusion.

i. "All hummingbirds are richly colored."

ii. "No large birds live on honey."

iii. "Birds that don't live on honey are dull in color."

iv. "Hummingbirds are small."

Express these statements using quantifiers (i.e. $\forall, \exists$), logical connectives (i.e. $\wedge, \vee, \sim, \Rightarrow, \Leftrightarrow$), and the following predicates:

- $P(x) :=$ "x is a hummingbird"
- $Q(x) :=$ "x is large"
- $R(x) :=$ "x lives on honey"
- $S(x) :=$ "x is richly colored"

For example, the statement "All hummingbirds live on honey" would be expressed by: $\forall x,\ P(x) \Rightarrow R(x)$.

(b) Show that the conclusion is valid.

4. Dynamic Programming [$5 + 10 + 5 = 20$ points]
Given two strings $x = x_1 x_2 \cdots x_m$ and $y = y_1 y_2 \cdots y_n$, we wish to find the length of their longest common substring. That is, the largest k such that there are indicies i and j such that $x_i x_{i+1} \cdots x_{i+k-1} = y_j y_{j+1} \cdots y_{j+k-1}$.

   (a) Let $L(i, j)$ be the length of the longest common substring between $x_1 \cdots x_i$ and $y_1 \cdots y_j$ that ends at x_i and y_j. Write a recurrence describing the relationship between $L(i, j)$ and $L(i-1, j-1)$.

   (b) Write a dynamic programming algorithm to find the length of the longest common substring between x and y. Be sure to include all initializations and show that the algorithms runs in time $O(mn)$.

(c) How might you modify your algorithm so that the space complexity is $O(\min\{m, n\})$? Pseudocode is not necessary, but explain your solution in detail.

5. Graph algorithms [15 + 5 = 20 points]

(a) You are given an undirected graph $G = (V, E)$ with edge weights $w(e)$ for every $e \in E$. You are also given a minimum spanning tree T of G. Let $e' \in E$ be some edge in G. Describe an $O(|E|)$ algorithm to find the minimum spanning tree of the graph after removing e', that is, of the graph $G' = (V, E \setminus \{e'\})$.

(b) During the execution of a Breadth-First-Search (BFS) in an unweighted, undirected graph G starting from a node S, consider the case when two vertices A and B are both present in the queue at some point in time. Let $\text{dist}(x, y)$ be the graph distance between nodes x and y in the graph G. What are the possible value of $\text{dist}(S, A) - \text{dist}(S, B)$? Be sure to justify your answer.

6. Divide and Conquer [20 points]

Given an (unsorted) array of numbers, a local maximum is at index i if $A[i]$ is larger than $A[i-1]$ and $A[i+1]$. (The last and first elements only need be larger than the previous and next elements respectively.) Give an algorithm with running time $O(\log n)$ to find a local maximum in an array of length n where no entries are repeated.

Note: A brief but precise description of the algorithm is required. Pseudocode is not required, but may be provided if you think it helps your exposition. Do not forget to provide the running time analysis of your algorithm.