COSC 302, Practice Exam #2
April 3, 2018

Honor Code

I agree to comply with the spirit and the rule of the Colgate University Academic Honor Code during this exam. I will not discuss the contents of this exam with other students until all students have finished the exam. I further affirm that I have neither given nor received inappropriate aid on this exam.

Signature: ______________________________

Printed Name: ______________________________

Write and sign your name to accept the honor pledge. Do not open the exam until instructed to do so.

You have 50 minutes to complete this exam.

There are 5 questions and a total of 68 points available for this exam. Don't spend too much time on any one question.

If you want partial credit, show as much of your work and thought process as possible.

Question	Points	Score
1	6	
2	12	
3	15	
4	15	
5	20	
Total:	68	

1. (6 points) Answer the following True/False questions by clearly circling your answer. No justification is required.

   (a) **True or False**: In a red-black tree storing n elements, every simple path from the root to a descendant leaf has $\Omega(\lg n)$ nodes.

   (b) **True or False**: The worst-case time to search for an element in a hash table with open addressing is $\Theta(1)$.

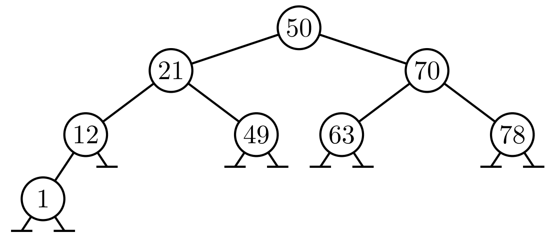
   (c) **True or False**: In a graph $G = (V, E)$, a simple path cannot have more than $|V| - 1$ edges.

   (d) **True or False**: The Bellman-Ford algorithm always runs in $\Theta(VE)$ time.

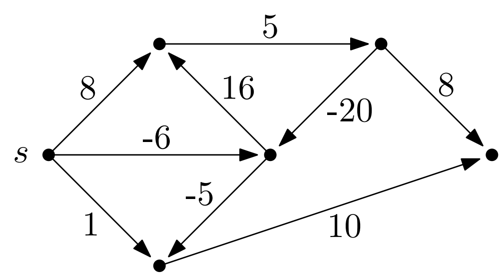
   (e) **True or False**: Every graph has a topological ordering.

   (f) **True or False**: Breadth-first search on an undirected graph yields only tree edges and cross edges.

2. (a) (6 points) Consider the following binary search tree. Can its nodes be colored so that it is a valid red-black tree? If so, color the red nodes. If not, explain why not.



(b) (6 points) Highlight a shortest-paths tree with source vertex s on the following graph.



3. (15 points) Mr. Skeptical questions the value of implementing a priority queue with a max-heap data structure. In particular, he believes a balanced binary search tree (BBST), such as a red-black tree, can do everything a max-heap can do with the same running time. You decide to implement several operations to see if he is right.

   (a) Briefly describe how to implement EXTRACT-MAX with a BBST. Give as tight as possible best- and worst-case bounds on the running time of your algorithm.

   (b) Briefly describe how to efficiently build the proposed priority queue data structure given an input array of elements with their priorities. Give as tight as possible the best- and worst-case bounds on the running time of your algorithm.

   (c) Is Mr. Skeptical right? Circle one: **Yes / No** Explain your answer.

4. (15 points) In this problem we will solve the minimum spanning tree problem with edge weight constraints. Suppose you are given a connected, undirected, weighted graph $G = (V, E, w)$.

   (a) Suppose that all edge weights are 3. Describe an algorithm to compute a minimum spanning tree in G in $O(V + E)$ time. Argue for, but do not formally prove, correctness of your algorithm.

   (b) Suppose that all edge weights are now 1 or 0. Describe an algorithm to compute a minimum spanning tree in G in $O(V+E)$ time. Formally prove that your algorithm is correct.

5. (20 points) It's snowing outside and you and a friend are bored. You decide to make up a game with building blocks. Your friend gives you a box of k blocks of various positive integer heights, and challenges you to build a tower of height n (which is also a positive integer) with some subset of your blocks. Can you build it? You decide to solve this problem with dynamic programming.

   (a) Suppose it *is* possible to build a tower of exactly height n using a subset of your k blocks. Define the subproblems that you would use to solve this problem efficiently with dynamic programming. Give a recurrence that can be used to solve this problem.

   (b) Sketch the array that you would use to solve this problem with dynamic programming and draw arrows to indicate an order in which the array should be filled. Note the array's dimensions, and mark where the final solution is stored.

Problem continued on next page →

(c) Now suppose that it might not be possible to build a tower of height n. How can you efficiently determine whether or not it is possible?

(d) Describe how to change your algorithm to efficiently determine the largest height $h \leq n$ of a tower that can be built with the k blocks.

(e) Now describe how to change your algorithm to efficiently determine the smallest height $h \geq n$ of a tower that can be built with the k blocks.

(This page is intentionally blank. Label any work with the corresponding problem number.)

(This page is intentionally blank. Label any work with the corresponding problem number.)