CS4231: Analysis of Algorithms I, Fall 2018

Final Exam, December 6, 2018

This exam lasts 1 hour and 15 minutes. It contains 5 problems, some of them composed of several parts. There are 100 points in all, and you have 75 minutes. Do not spend too much time on any problem. Read them all through first and attack them in the order that allows you to make the most progress.

The exam is closed book, notes, and electronics. You may use two 8.5”x11” crib sheets (both sides).

You can use any algorithm that we covered in class or the homeworks by simply referring to it and specifying the input. You do not need to repeat the algorithms.

Write your solutions in the space provided (and only there).
You can use the blank pages at the end of this booklet for scratch space.
Be *clear, precise and succinct* in your answers. You will be graded not only on the correctness of your answers, but also on the clarity with which you express them.
Be neat.

Good luck!

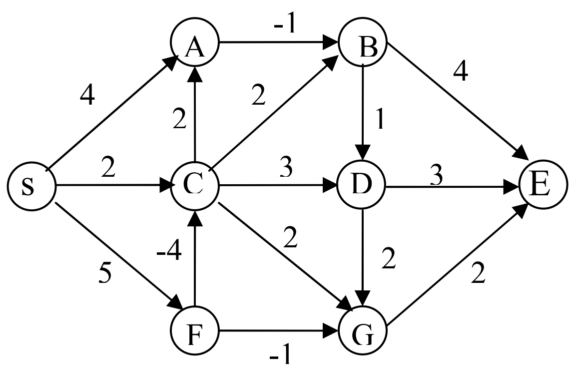
NAME (Last, First): ____

UNI: ________________________________

Problem	Max Points	Points
1	20	
2	30	
3	15	
4	15	
5	20	
Total	100	

Problem 1 [20 points]

1. [10 points]   Consider the following weighted directed graph with the edges weighted as indicated in the figure. (Note, some of the weights are negative.)



Write down in the following table the shortest (minimum weight) distances from node s to all the other nodes, and draw below it a shortest path tree from s. You do not need to provide any justifications.

A	B	C	D	E	F	G

Shortest path tree:

2. [10 points] Give an optimal Huffman code for the alphabet {a,b,c,d,e,f,g} with the following frequencies: a:14, b:3, c:7, d:5, e:10, f:1, g:2. Show the code in terms of the corresponding optimal tree. You do not need to provide any justification.

Problem 2 [30 points, 6 points per part]

For each of the claims below circle True or False, and justify your answer. *Your justifications are more important than the True/False designations, and must be* ***succinct, precise and convincing****.*
All the parts can be adequately answered and justified in a few lines.

All the times below refer to worst-case time.

a. Recall that a graph G has in general many adjacency list representations because the adjacency list of each node contains the adjacent nodes in arbitrary order. Suppose that we perform Breadth First Search of an undirected graph from a given source node s.

Claim: The BFS tree does not depend on the adjacency list representation of the graph, i.e. all representations yield the same BFS tree.

True False

b. We are given a weighted undirected graph *G* with positive integer weights on the edges, and a minimum spanning tree *T*.

Claim: If we square the weights of all the edges, then *T* is still a minimum spanning tree of the graph with the new weights.

True False

c. We are given a directed graph G=(N,E) with node set N={1,…,n} and edge set E, by its adjacency list representation.

Claim: We can compute all cycles of length 2 of G in O(|N|+|E|) time and space.

True   False

d. We are given a directed graph $G=(N,E)$ and two disjoint subsets U, V of the nodes. We want to find a minimum-size subset F of edges whose deletion disconnects U from V, i.e., a subset F with as few edges as possible such that the graph $G\text{-}F = (N,E\text{-}F)$ does not contain any path from any node of U to any node of V.

Claim: This problem can be solved in polynomial time.

True False

e. *Claim.* If problems A and B are NP-complete and A is in P then B is also in P.

True False

Ungraded Scratch Space

Problem 3. [15 points]

We are given a directed graph $G=(N,E)$ by its adjacency list representation, and a partition of its nodes into red and blue nodes, i.e., sets R, B, where $R \cup B = N$ and $R \cap B = \emptyset$. We want to determine whether there exists a simple cycle in G that contains both a red node and a blue node, i.e. a simple cycle C such that $C \cap R \neq \emptyset$ and $C \cap B \neq \emptyset$.

Give an algorithm that solves this problem in time $O(|N|+|E|)$. Justify the correctness of your algorithm and the running time.

If you cannot achieve linear time, give the most efficient algorithm you can, for partial credit, and state explicitly its running time.

Ungraded Scratch Space

Problem 4. [15 points]

We are given n points $x_1 < x_2 < \ldots < x_n$ on the real line in sorted order, and a positive integer $k \leq n$. We want to find k closed intervals whose union contains the n given points $x_1, x_2, \ldots, x_n$, such that the sum of the lengths of the intervals is as small as possible. For example, if k=1, the optimal solution is the interval $[x_1, x_n]$; if k=n, the optimal solution consists of the n trivial intervals $[x_i, x_i]$, i=1,…,n.

Give an algorithm that solves this problem in O(n) time. Justify the correctness of your algorithm and the running time. (Note that k is part of the input, not a constant.)

If you cannot achieve O(n) time, give the most efficient algorithm you can, for partial credit, and state explicitly its running time.

(Hint: What is the optimal solution if k=2? If k=3?)

Problem 5. [20 points]

Given an unlimited supply of coins of denominations c_1, c_2,…, c_n, we wish to make change for a given value V, if possible; that is, we wish to find a set of coins whose total value is V, where we are allowed to use multiple coins of the same denomination if we wish. It might not be possible in general to make change for V; for example if the denominations are 5, 8 then we can make change for 18 (e.g. 18=5+5+8), but we cannot make change for 14.

Give an algorithm, which takes as input positive integers c_1, c_2,…, c_n, and V and solves the above coin-changing problem in $O(nV)$ time. The algorithm should either determine that it is impossible to make change for V, or it should output a set of coins whose sum is equal to V. Justify the correctness and running time of your algorithm.

(Hint: Use dynamic programming.)

Ungraded Scratch Space

Ungraded Scratch Space

Ungraded Scratch Space

Ungraded Scratch Space

Ungraded Scratch Space