University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2011

Name:	
Email:	

- "Nothing is true. All is permitted" - Friedrich Nietzsche. Well, not exactly. *You are not permitted to discuss this exam with any other person.* If you do so, you will surely be smitten. You may consult any *other* sources including books, papers, web pages, computational devices, animal entrails, seraphim, cherubim, etc. in your quest for truth and solutions. Please acknowledge your sources.
- *Show your work!* You will not get full credit if we cannot figure out how you arrived at your answer. A numerical solution obtained via a computer program is unlikely to get much credit, if any, without a correct mathematical derivation.
- Write your solution in the space provided for the corresponding problem.
- If any question is unclear, ask for clarification.

Question	Points	Score	Grader
1	10		
2	10		
3	20		
4	20		
5	20		
6	20		
7	30		
Total	130		

1. **Spanning Trees**

   (a) (5 points) Prove that in a connected graph where all edge weights are the same, that any breadth first search tree will also be a minimum spanning tree.

   (b) (5 points) Consider an undirected graph where every edge has a unique positive weight. Is it the case that any shortest path tree rooted at v on that graph is always the same as the minimum spanning tree found by Prim's algorithm when seeded initially with the vertex v. If so, prove it. If not, give a counter example.

2. **Donuts** A certain bakery sells donuts in boxes of three different quantities, x_1, x_2, and x_3. In the Donut Buying problem, you are given the numbers x_1, x_2 and x_3, and an integer n and you should return either 1) the minimum number of boxes needed to obtain exactly n donuts if this is possible, along with a set of boxes that obtains this minimum; or 2) "DOH!" if it is not possible to obtain exactly n donuts.

   For example if $x_1 = 4$, $x_2 = 6$, $x_3 = 9$ and $n = 17$, then you should return that 3 boxes suffices, with 2 boxes of size 4, and 1 box of size 9. However, if $n = 11$, you should return DOH! since it is not possible to buy exactly 11 donuts with these box sizes.

   (a) (5 points) For any positive x, let $m(x)$ be the minimum number of boxes needed to buy x donuts if this is possible, or INFINITY otherwise. Write a recurrence relation for the value of $m(x)$. Don't forget the base case(s)!

   (b) (5 points) Give an efficient algorithm for solving Donut Buying. How does its running time depend on x_1, x_2, x_3, and n? Is it an algorithm that runs in polynomial time in the input sizes?

3. **Amortized Analysis** (20 points) Consider a new data structure that combines the properties of both stacks and queues. It may be viewed as a list of elements written left to right with three possible operations:

   - Push: add a new item to the left end of the list
   - Pop: remove the item on the left end of the list
   - Pull: remove the item on the right end of the list

   Show how to implement this new data structure using only: one stack, one queue, and $O(1)$ additional memory, so that the amortized time for all three operations is $O(1)$. **You are allowed to access the stack and queue only through the standard operations: PUSH and POP for the stack, and PUSH and PULL for the queue.**. Don't forget to prove your operations have $O(1)$ amortized cost.

4. **Commodities Trading** (20 points) Imagine there are n commodities $g_1, g_2, \ldots, g_n$ and an n by n table R of exchange rates, such that one unit of commodity g_i can be traded for $R[i, j]$ units of commodity j. Moreover, there are *taxes* on each commodity given by $t_1, t_2, ..., t_n$ such that if i is an intermediate commodity in a sequence of trades, you are taxed at rate t_i when you convert to commodity i. For example, when you convert to the intermediate commodity "pork bellies", you are taxed at a rate of .05. A *valid* sequence of trades, starts and ends with the same commodity. The *revenue* from a sequence of trades is the amount of the first commodity you end up with if you start with one unit of the first commodity initially and perform the trades in the sequence. For example, if there is a sequence of commodities $g_{i_1}, g_{i_2}, \ldots, g_{i_k}, g_{i_1}$, then the profit for that sequence is: $R[i_1, i_2] \cdot (1 - t_2) \cdot R[i_2, i_3] \cdot (1 - t_3) \cdots R[i_{k-1}, i_k] \cdot (1 - t_k) \cdot R[i_k, i_1] \cdot (1 - t_1)$. Note that you are only taxed once for the start/end commodity in the valid sequence. Note also that the revenue may be less than 1.

   Your goal is to design an algorithm that finds some valid sequence of trades with revenue greater than 1 if such a sequence exists, or outputs NONE if no such sequence exists. Show how to do this using techniques from class. Don't forget to analyze the runtime of your algorithm.

5. **Dinner Party** (20 points) You are giving a large dinner party. There are n guests who you will have to split up over two tables. Assume that for any two people, i and j, there is a nonnegative cost $c_{i,j}$ if they sit at different tables. Moreover, each person has given you information about their table preferences: for each i there is a nonnegative cost a_i if they do not sit at table A, and a nonnegative cost b_i if they do not sit at table B. Your goal is to divide the guests into two sets A and B, seating them at the corresponding tables, in a way that minimizes the total cost: $\sum_{i \in A} b_i + \sum_{i \in B} a_i + \sum_{(i,j): i \in A \text{ and } j \in B} c_{i,j}$

   Give an efficient algorithm to solve this problem based on min-cuts. Argue that your algorithm is correct and analyze its runtime.

6. **Rock, Paper, Scissors**

   Rock, Paper, Scissors is a simple 2 person game. In a given round, both players simultaneously choose either Rock, Paper or Scissors. If they both choose the same object, it's a tie. Otherwise, Rock beats Scissors; Scissors beats Paper; and Paper beats Rock. Imagine you're playing the following betting variant of this game with a friend. When Scissors beats Paper, or Paper beats Rock, the loser gives the winner \$1. However, in the case when Rock beats Scissors, this is called a SMASH, and the loser must give the winner \$10.

   (a) (6 points) Say you know that your friend will choose Rock, Scissors or Paper, each with probability 1/3. Write a linear program to calculate the probabilities you should use of choosing each object in order to maximize your expected winnings. Let p_1, p_2, p_3 be variables associated with the best way of choosing Rock, Scissors and Paper respectively. Note: If you want to check your work, there are several free linear program solvers on the Internets: check the Wikipedea page on linear programming.

(b) (14 points) Now say that your friend is smart and, also, clairvoyant: she will magically know the exact probabilities you are using and will respond optimally. Write another linear program to calculate the probabilities you should now use in order to maximize your expected winnings. Hint 1: If your opponent knows your strategy, her strategy will be to choose one of the three objects with probability 1. Hint 2: Review the LP for shortest paths in the last HW.

7. **File Movement**

In the File Movement problem, you want to move files from an initial configuration to a goal configuration in a network of n completely connected computers, as quickly as possible. The files are all of equal size, and in any one round, a computer can be involved in either sending a file or receiving a file, **but not both** [1]. You've decided that you can recast this problem as a problem in graph theory as follows. You are given a graph $G = (V, E)$, where V is the set of all computers, and there is an edge from x to y in E for each file that starts at x and wants to go to y. **Note that there may be multiple edges in G from x to y if multiple files want to move from x to y.** Your goal is to label each edge in G with a positive integer such that: 1) for every node v in G, all edges touching v have unique numbers; and 2) the largest integer used is as small as possible.

See the figure below for an example.

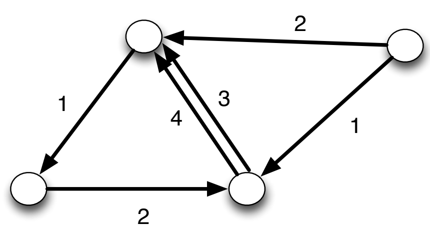

Figure 1: File Movement problem with 4 computers and 6 files. The labeling shows that all files can be moved in 4 rounds.

(a) (10 points) Assume you are given a directed graph G where every vertex has out-degree k and in-degree k, for some value $k \geq 2$. Describe an algorithm, based on techniques from this class, that gives a valid labeling for G, and that uses no more than $3k$ different numbers. Show that your algorithm is correct and analyze its runtime when n is the number of vertices in G. Hint 1: Repeatedly find a collection of cycles such that every vertex in the graph appears exactly once, in exactly one cycle in the collection. Hint 2: bipartite matching.

[1] The files are so large that they must be read or written to disk directly. Thus, either sending or receiving a file requires complete control of the disk.

(b) (10 points) Now assume that you are given directed graph G over n vertices, and also have an additional $n/3$ free "helper" computers. Show how you can use these helper computers to get a labeling that requires only $2k$ numbers.

(c) (10 points) **(Challenge)** Now what if the maximum degree of G is K, but G is not necessarily regular. Describe an efficient algorithm to find a labeling that uses at most $\lceil (3/2)K \rceil$ numbers without any helper computers, and at most $K + 1$ number with $n/3$ helpers. Hint: Google "Eulerian Cycle". Assume you have an efficient algorithm to solve Eulerian Cycle.