ICS 260 – Fall 2001 – Final Exam

Name:

Student ID:

1:

2:

3:

4:

5:

6:

Total:

1. Matrix rounding. (25 points)

Suppose we are given the following 3×3 matrix as input (shown also with the sums of each row and of each column).

63	14	84	161
71	53	38	162
21	94	49	164
155	161	171	

We wish to output a rounded form of the matrix, where each entry is rounded up or down to the nearest multiple of ten, as are the row and column sums. The rounded sums should still be an accurate sum of the row and column, however, so e.g. it would not be allowed to round every entry in the first row of the matrix downwards because that would cause the sum of the rounded row to differ from the rounded sum of the row.

Draw a graph, with upper and lower bounds on the flow amounts for each edge, such that a feasible circulation in this graph corresponds to a solution to the matrix rounding problem. You do not need to solve the feasible circulation problem in your graph, and you do not need to find a rounded form of the matrix.

2. Change-making problem. (30 points)

Suppose you are making change for n cents with three types of coins, worth 5, 10, and 25 cents (e.g. the US nickel, dime, and quarter). For example, there are six different ways of making change for $n = 35$: 25+10, 25+5+5, 10+10+10+5, 10+10+5+5+5, 10+5+5+5+5+5, and 5+5+5+5+5+5+5. You may assume that n is a multiple of five.

Describe a general technique for transforming such a change-making problem into a directed acyclic graph, with two of the graph vertices labeled as s and t, in such a way that the paths from s to t in the graph correspond to the different ways of making change for n cents. Also draw the graph resulting from using your technique on the example $n = 35$.

Your technique must produce graphs of size $O(n)$ for the input n. For full credit, no two paths in the graph should represent the same way of making change (so your example with $n = 35$ should have exactly six paths from s to t). No credit will be given to techniques that work by listing all methods of making change and creating a separate path for each such method.

3. Huffman coding. (20 points)

Suppose we are given the following input to the greedy Huffman code construction algorithm:

symbol	frequency
a	12
b	5
c	3
d	9
e	4
f	6

Describe the Huffman code that would be constructed by the algorithm from this input. You may either show it as a tree, or write the sequence of binary digits used to encode each symbol.

4. PC-board drilling. (25 points)

Suppose you are managing a PC-board manufacturing factory. One of the steps each PC-board must undergo is drilling holes for components: most components are surface-mounted, but a few are placed in holes and then soldered. You must design a program that determines the sequence in which the holes are drilled in each board.

You may assume the following: It takes 1/2 second to drill a hole, and one second per inch to move the drill to each successive hole. Each board needs roughly 20 holes, and there are typically 200 identical boards to be drilled in a job lot. You are willing to dedicate a powerful PC to the drill sequencing task, and it can work on finding the sequence for one job lot while the previous job lot is being drilled. The drilling step is the bottleneck for your factory, so the faster you can perform it the more business you can handle, but you can not afford to buy more than one drill.

Would it be more appropriate to solve this drill sequencing problem using Christofides' heuristic, or by branch-and-bound? Explain your answer.

5. Complexity theory. (10 points)

For each of the following sentences, state whether the sentence is known to be true, known to be false, or whether its truth value is still unknown.

(a) If a problem is in P, it must also be in NP.

(b) If a problem is in NP, it must also be in P.

(c) If a problem is NP-complete, it must also be in NP.

(d) If a problem is NP-complete, it must not be in P.

(e) If a problem is not in P, it must be NP-complete.

6. NP-completeness (40 points). If you get at least 30 points on this question, you will be guaranteed at least a B+ for your overall course grade.

In a certain town, there are many clubs, and every adult belongs to at least one club. The townspeople would like to simplify their social life by disbanding as many clubs as possible, but they want to make sure that afterwards everyone will still belong to at least one club.

Prove that the Redundant Clubs problem is NP-complete. You may make use of the known NP-completeness of the Maximum Independent Set, Set Cover, or Traveling Salesman Problems (all described below).

PROBLEM NAME: Redundant Clubs
INPUT: List of people; list of clubs; list of members of each club; number K.
OUTPUT: Yes if there exists a set of K clubs such that, after disbanding all clubs in this set, each person still belongs to at least one club. No otherwise.

PROBLEM NAME: Maximum Independent Set
INPUT: Undirected graph G and number K.
OUTPUT: Yes if there exists a set of K vertices in G such that no pair of vertices in the set is connected by an edge. No otherwise.

PROBLEM NAME: Set Cover
INPUT: List L of elements, family F of subsets of elements, number K.
OUTPUT: Yes if there exist some K subsets in the family the union of which is all of L. No otherwise.

PROBLEM NAME: Traveling Salesman Problem
INPUT: Matrix of distances between cities, number K.
OUTPUT: Yes if there is a tour that visits each city once and has total length at most K, no otherwise.

You may use this page for scratch paper.

You may use this page for scratch paper.