ICS 260 – Fall 2001 – Second Midterm

Name: Answer Key

Student ID:

1: 20

2: 30

3: 20

4: 30

5: 20

Total:

1. Longest symmetric subsequence. (20 points)

A *symmetric sequence* or *palindrome* is a sequence of characters that is equal to its own reversal; e.g. the letters in the phrase “a man, a plan, a canal: panama!” form a palindrome.

Describe how to use the longest common subsequence algorithm described in class to find the longest symmetric subsequence of a given sequence.

LCS(string, reverse(string))

2. Change-making problem. (30 points)

The following pseudocode finds the minimum number of coins needed to make change for a given number, in an arbitrary coinage system. The number `k` represents the number of different coin types available, the array `C[i]` represents the list of different coin values in the coinage system, the number `n` represents the amount of money to be changed, and the array `M[j]` computed by the algorithm represents the number of coins needed to make change for `j` units of money.

```
M[0] = 0
for j in 1, 2, ... n:
    M[j] = infinity
    for i in 0, 1, ... k-1:
        M[j] = min(M[j], M[j-C[i]]+1)
```

After we perform this pseudocode, the desired number of coins will be in `M[n]`, but we may want to know the actual set of coins needed to make change for `n` units of money.

Write pseudocode that efficiently computes a set of coins which makes change for `n` units of money using as few coins as possible. You should assume that the array `M[j]` has already been computed using the pseudocode above. Your pseudocode should store integer values into an array `X[i]`, representing the number of coins of type `i` used by your set of coins.

```
for i in 0, 1, ... k-1:
    X[i] = 0

while n > 0:
    find i such that M[n] == M[n-C[i]]+1
    X[i] = X[i] + 1
    n = n - C[i]
```

3. Invasion percolation. (20 points)

Earth scientists use the following technique to model the seepage of water into soil, the erosion of drainage basins, and similar phenomena: begin with a grid of cells, each of which is initially dry, and set the strength of the wall between each pair of adjacent cells to a random number. Mark a single cell (the source of the seepage or the mouth of the drainage basin) as being wet. Then, repeatedly find the weakest wall that connects any wet cell to any dry cell and break that wall, making the dry cell wet.

(a) Which of the following algorithms would be the best choice to determine the sequence of wall-breaking events performed in this invasion percolation model? Circle your answer.

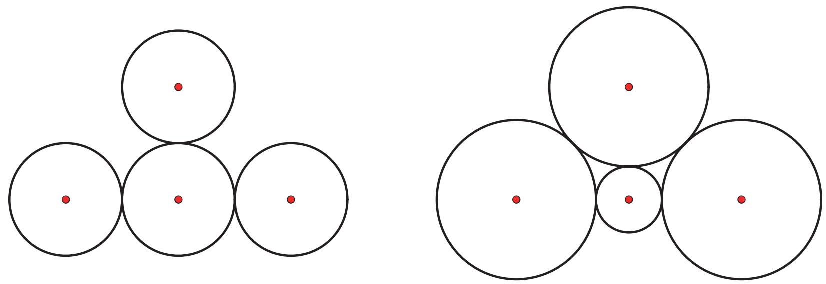
- Boruvka's algorithm
- Christofides' algorithm
- Ford-Fulkerson algorithm
- Huffman tree algorithm
- Hybrid MST algorithm
- Optimal binary search tree algorithm
- Prim-Dijkstra algorithm
- Simplex algorithm

(b) What would be the time for performing the algorithm you chose for a two-dimensional square $n \times n$ grid of cells? Use O-notation.

$O(n^2 \log n)$

4. Circle packing. (30 points)

Suppose we are given as input four points in the plane, A, B, C, and D. We wish to place four circles, centered at each of these points, so that no two circles overlap each other and the average radius is as large as possible. The figure below depicts two possible solutions for the points (0,0), (1,0), (-1,0), and (0,1); the right one has larger average radius.



(a) Describe how to set up this average-radius-maximization problem as a linear program, in which the variables a, b, c, d represent the radii of the four circles. You do not need to use standard form or slack form, but do write out the objective function and inequalities used for your linear program. You may use $d(p,q)$ to denote the distance between points p and q.

$$\text{maximize } a+b+c+d$$

$$a \geq 0, b \geq 0, c \geq 0, d \geq 0$$

$$\begin{aligned} a+b &\leq d(A,B) \\ a+c &\leq d(A,C) \\ a+d &\leq d(A,D) \\ b+c &\leq d(B,C) \\ b+d &\leq d(B,D) \\ c+d &\leq d(C,D) \end{aligned}$$

(b) Let $NN(x)$ denote the distance from x to its nearest neighbor. Observe that, in any feasible solution to the circle packing problem, each radius is at most $NN(x)$. On the other hand, setting each radius to $NN(x)/2$ (as in the left figure) produces a solution in which no two circles overlap. What can you say about the approximation ratio of this solution?

$$\sum NN(x)/2 \leq a+b+c+d \leq \sum NN(x)$$

so the approximation ratio is at least 1/2 (I would also give full credit if you said it was at most 2).

This problem generalizes to any number of points larger than four in any metric space. I don't know of a good combinatorial (non-LP-based) algorithm; it might be interesting to understand the structure of the graphs formed by pairs of tangent circles in the optimal solution. There are point sets in general metric spaces for which the approximation ratio of the $NN(x)/2$ solution is arbitrarily close to 1/2, but it might be possible to prove a better approximation ratio in the plane.

5. Simplex method. (20 points)

Suppose we are solving the following linear program (in slack form):

$$\text{maximize } x_1 - 2x_2 - x_3$$
$$x_i \geq 0 \text{ for } 1 \leq i \leq 6$$

$$\begin{aligned} x_4 &= 1 + x_2 - x_3 \\ x_5 &= 2 - x_1 + x_2 \\ x_6 &= 3 - 3x_1 + x_3 \end{aligned}$$

(a) What is the basic solution associated with this slack form?

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 0 \\ x_3 &= 0 \\ x_4 &= 1 \\ x_5 &= 2 \\ x_6 &= 3 \end{aligned}$$

(b) Describe the new basic solution resulting from a single step of the simplex method. You do not need to describe the new slack form corresponding to the new basic solution.

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 0 \\ x_3 &= 0 \\ x_4 &= 1 \\ x_5 &= 1 \\ x_6 &= 0 \end{aligned}$$