CSE 7350/5350
D.W. Matula

Homework Set # 5
Dynamic Programming

Total Points: **50**

Due date:
04 December 2013

1. [De-merging]

If two sequences $a_1, a_2, \ldots, a_m$ and $b_1, b_2, \ldots, b_n$ are interleaved, we say that the resulting sequence $c_1, c_2, \ldots, c_{m+n}$ is a *shuffle* of the first two. For example,

ABBAADCCDBABACD

is a shuffle of ABADCBAC and BACDABD since it can be obtained by interleaving those two sequences in this way:

```
AB   ADC    B    AC
  BA     CD   AB    D
```

You are to give a dynamic programming algorithm for determining whether or not a given sequence is a shuffle of two other given sequences. Your algorithm is to run in time O(mn), where m, n and $m + n$ are the lengths of the three sequences. You should carefully describe each step of the process, e.g. this should include certain definitions, the principal recurrence relations, the table setup, the order in which the table contents are computed, and the computation storage window utilized.

2. [Shortest Path Count]

a) Describe a dynamic program that determines the distance matrix and counts the number of distinct shortest paths between each pair of vertices of a graph that runs in time O(|V||E|) given the adjacency list for the graph.

b) Apply your algorithm to the 16 vertex graph illustrated in the dual MAS walkthrough in the Maximum Adjacency Search Project Description on the homepage. (If done by hand show only the 4x16 portion for rows a, b, c, d).

c) Discuss how you can use the results of (a) to find an edge of the graph that has the highest count of distinct shortest paths including that edge.

d) (5 point bonus) Find an edge as described in part (c) for the 16 vertex graph of part (b).

3. [Longest palindrome subsequence]
A palindrome is a nonempty string over some alphabet that reads the same forward and backward, e.g. racecar. Give an efficient algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input CHARACTER your algorithm should return CARAC . What is the running time of your algorithm?

How much space do you need to simply determine the length of a longest palindrome subsequence? [Hint: see text discussion of longest common subsequence.]

Illustrate your algorithm on the following sequence.

BFADFBECDAEBFCACEFDBAEDC.

4. [Monotonic Subsequences]
Given the sequence a(1), a(2),..., a(n), of real numbers, consider the problem of determining the length *k* of a longest monotonically decreasing subsequence $a(i_1) > a(i_2) > \ldots > a(i_k)$ where $i_1 < i_2 < \ldots < i_k$.

Using the paradigm of dynamic programming give the best algorithm you can for determining a longest monotonically decreasing subsequence. Describe the algorithm and analyze its time and space complexity. (See text p. 319, 16.3-5, 6). Discuss the storage space required to solve for the variable *k* compared to obtaining a longest sequence.

Apply your algorithm to the following data, forwards and backwards:

54, 76, 30, 44, 74, 15, 78, 67, 36, 46, 11, 77, 42, 49, 82, 73, 80,
66, 52, 58, 22, 68, 35, 40, 24, 13, 55, 27, 39, 16, 43, 93, 61, 53,
94, 49, 74, 45, 60, 83, 18, 73, 42, 69, 67, 22, 61, 30, 63, 51, 62.

CSE 7350/5350
D.W. Matula

Homework Set # 5
Dynamic Programming
Total Points: **50**

Due date:
04 December 2013

5. [Randomized Dynamic Subsequence Selection]

a) Consider the dynamic ascending subsequence selection problem as illustrated in the SOLO game discussed in class.

Write a program to compute the expected value of the 52 card game. Tabulate the solution of the n card game and the "take point" for $n=2^k$, where $1 <= k <= 18$. Estimate functions describing the general solution for the value and take point for arbitrary n.

b) (5 points bonus) How little space is sufficient for computing the value of the n card game?