Name:____________________________ PID:______________________

COT5405 - Homework 3

Out date: **10/22/2010 (Friday)**, due date: **11/03/2010 (Wednesday)**
15 points each problem.
You need to turn in the solutions for **all eight** problems. But we will select **four** problems and **only** grade these four.

Let us take the longest increasing subsequence problem as an example for dynamic programming.

Input: a sequence of numbers $a_1, a_2, \ldots, a_n$. A subsequence is any subset of these numbers taken in order, of the form $a_{i1},\ldots,a_{ik}$, where 1<=i1<i2<...<ik<=n, and an increasing subsequence is one in which the numbers are getting strictly larger.
Output: the increasing subsequence of greatest length.

Let L[j] be the length of the longest increasing subsequence ending exactly at a_j.

If j==1

L[j] = ______1____________________

else if 1<=j<=n

$$L[j] = 1 + \max_{1\le i<j \text{ and } a_i<a_j} \{L[i]\}$$

____

Output:

$$\max_{1\le j\le n} \{L[j]\}$$

____

Name:__________________________ PID:_____________________

6.1. A *contiguous subsequence* of a list S is a subsequence made up of consecutive elements of S. For instance, if S is

$$5, 15, -30, 10, -5, 40, 10,$$

then $15, -30, 10$ is a contiguous subsequence but $5, 15, 40$ is not. Give a linear-time algorithm for the following task:

Input: A list of numbers, $a_1, a_2, \ldots, a_n$.
Output: The contiguous subsequence of maximum sum (a subsequence of length zero has sum zero).

For the preceding example, the answer would be $10, -5, 40, 10$, with a sum of 55.

(*Hint:* For each $j \in \{1, 2, \ldots, n\}$, consider contiguous subsequences ending exactly at position j.)

Note: Do not output the consecutive subsequences; output the maximum sum.

Let S[j] be the maximum sum of all consecutive subsequences ending exactly at a_j.
If j==0

S[j] = ____________________

else if j >=1 and j <= n

S[j] =

____

Output =

____

Name: ______________________________ PID: ______________________

6.2. You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \cdots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance a_n), which is your destination.

You'd ideally like to travel 200 miles a day, but this may not be possible (depending on the spacing of the hotels). If you travel x miles during a day, the *penalty* for that day is $(200 - x)^2$. You want to plan your trip so as to minimize the total penalty—that is, the sum, over all travel days, of the daily penalties. Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.

Note: Output the minimum total penalty.

Let S[j] be the minimum total penalty when you stop at hotel j.

If j==0

S[j] = ________________________

else if j >=1 and j <= n

S[j] =

____

Output = ________________________

Name: ____________________________ PID: ______________________

6.3. Yuckdonald's is considering opening a series of restaurants along Quaint Valley Highway (QVH). The n possible locations are along a straight line, and the distances of these locations from the start of QVH are, in miles and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are as follows:

- At each location, Yuckdonald's may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i, where $p_i > 0$ and $i = 1, 2, \ldots, n$.
- Any two restaurants should be at least k miles apart, where k is a positive integer.

Give an efficient algorithm to compute the maximum expected total profit subject to the given constraints.

Note: Output the maximum expected total profit.

Let F[j] be the total profit of restaurants that are within m_j miles from the start of QVH (i.e. at locations $m_1, m_2, \ldots, m_j$).
Let a(j) be index of the nearest location that is at least k miles behind location j (i.e. $(m_j - m_{a(j)})$ is greater than or equal to k and $(m_j - m_{a(j)+1})$ is less than k.).

If j==0

F[j] = ____________________

else if 1<=j<= n

F[j] =

____

Output = ____________________

Name: ___________________________ PID: ____________________

6.7. A subsequence is *palindromic* if it is the same whether read left to right or right to left. For instance, the sequence

$$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

has many palindromic subsequences, including A, C, G, C, A and A, A, A, A (on the other hand, the subsequence A, C, T is *not* palindromic). Devise an algorithm that takes a sequence $x[1 \ldots n]$ and returns the (length of the) longest palindromic subsequence. Its running time should be $O(n^2)$.

Note: Output the length of the maximum palindrome in the string.

Let F[i,j] denote the maximum palindrome subsequence in the string x[i,...,j].

if i>j

F[i,j] = __________________

else if i==j

F[i,j] = __________________

else if i<j

F[i,j] =

____

Output = __________________

Name: ____________________________ PID: _____________________

6.10. *Counting heads.* Given integers n and k, along with $p_1, \ldots, p_n \in [0, 1]$, you want to determine the probability of obtaining exactly k heads when n biased coins are tossed independently at random, where p_i is the probability that the ith coin comes up heads. Give an $O(nk)$ algorithm for this task.[2] Assume you can multiply and add two numbers in $[0, 1]$ in $O(1)$ time.

Let S[i,j] be the probability of tossing coins 1,2,...,i and obtaining exactly j heads.

If i==0
    if j==0
        S[0][0] = ___________________
    else j!=0
        S[0][j] = ___________________

else if 1<=i<=n

    S[i,j] =
        ____

Output =
        ____

Name: ____________________________ PID: ______________________

6.22. Give an $O(nt)$ algorithm for the following task.

Input: A list of n positive integers $a_1, a_2, \ldots, a_n$; a positive integer t.
Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once.)

Note: Do not output the subset; output either true or false.

Let S[i,j] indicate whether there exist some subset of $\{a_1,...,a_i\}$ such that the sum of elements in the subset is equivalent to j, where 0<=j<=t. S[i,j] can be true or false.

If i==0

    If j==0

        S[0,j] = ___________________

    else j!=0

        S[0,j] = ___________________

else if 1<=i<=n

    S[i,j] =

        ____

Output =

        ____

Name:___________________________ PID:_____________________

26.2-2

Show the execution of the Edmonds-Karp algorithm on the flow network below:

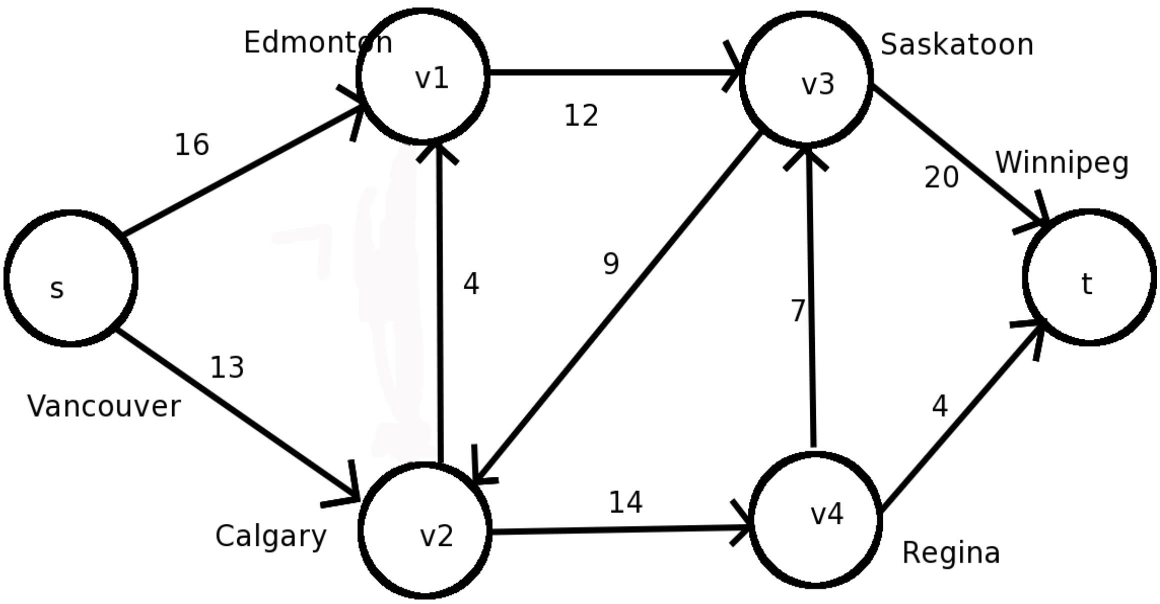

Figure: A flow network *G=(V,E)* for the Lucky Puck Company's trucking problem. The Vancouver factory is the source *s*, and the Winnipeg warehouse is the sink *t*. Pucks are shipped through intermediate cities, but only *c(u,v)* crates per day can go from city *u* to city *v*. Each edge is labelled with its capacity.

Name:___________________________ PID:_____________________

26.2-9

The edge connectivity of an undirected graph is the minimum number k of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how the edge connectivity of an undirected graph $G=(V,E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges.

Describe your idea clearly in a **short** paragraph.