CSCI 3104-Spring 2016: Assignment #6.

Assigned date: Wednesday, 3/9/2016,
Due date: Tuesday, 3/29/2016, before class
Maximum Points: 40 points (includes 5 points for legibility).

Note: This assignment *must be turned in on paper, before class.* Please do not email: it is very hard for us to keep track of email submissions. Further instructions are on the moodle page.

P1 (25 points) Write a dynamic programming algorithm to solve the longest increasing subsequence (LIS) problem. You are given an array a of n integers. An *increasing subsequence* is a sequence k indices

$$0 \leq i_1 < i_2 < \cdots < i_k \leq (n-1)$$

such that

$$a[i_1] < a[i_2] < \cdots < a[i_k]$$

Given a find the *longest increasing subsequence*. I.e, an increasing subsequence $i_1, \ldots, i_k$ whose length k is the longest possible.

Example: Input is $a : [1, 5, 2, 3, 8]$. The longest increasing subsequence has length 4. $i_1 = 0,\ i_2 = 2,\ i_3 = 3,\ i_4 = 4$ yielding the elements $1, 2, 3, 8$.

Let us define the function $\text{LIS}(a, j, M)$ as the length of the LIS for the first j elements of the array $(a[0] \cdots a[j-1])$, where j ranges from 0 to n and all elements of this subsequence are less than M.

(A) For the array $a : [1, 5, 2, 3, 8]$ write down the values of $\text{LIS}(a, 3, 2)$ and $\text{LIS}(a, 5, \infty)$.

(B) Write down the base cases for $\text{LIS}(a, 0, M)$.

(C) Write down a recurrence for $\text{LIS}(a, j, M)$ for any j with $1 \leq j \leq n$.

(D) Describe a bottom-up scheme to memoize the recurrence LIS above.

Describe the memo table itself: how many rows? how many columns? what do the rows and columns stand for? How are the base cases filled out in the table? How is each entry of the memo table filled out?

Also, show how the filled out memo table for the example array $a : [1, 5, 2, 3, 8]$.

(E, Extra Credit) Program your dynamic programming scheme in Python. Input to your function is the list a and output should be the LIS itself.

P2 (15 points) We are given a network below. Calculate the number of paths from node 0 to node 13.