Weekly Assignment 10:
Dynamic Programming 2

1. Determine a longest common subsequence of the strings $\langle 1,0,0,1,0,1,0,1\rangle$ and $\langle 0,1,0,1,1,0,1,1,0\rangle$, using the DP algorithm discussed during the lecture.

2. Write pseudocode for a bottom-up implementation of the dynamic programming algorithm for computing the longest common subsequence of two words $w = w_1 w_2 \cdots w_n$ and $v = v_1 v_2 \cdots v_m$. The output produced by your pseudocode should be a longest common subsequence for the two input words (not just its length). Add comments and explanations, allowing the TAs to understand your solution.

3. Let $G = (V, E)$ be a directed graph with nodes $v_1, \ldots, v_n$. We say that G is a *line-graph* if it has the following properties:

   1. Each edge goes from a node with a lower index to a node with a higher index. That is, every directed edge has the form (v_i, v_j) with $i < j$.
   2. Each node except v_n has at least one edge leaving it. That is, for every node v_i, $i = 1, 2, \ldots, n-1$, there is at least one edge of the form (v_i, v_j).

   The length of a path is the number of edges in it. The goal in this question is to solve the following algorithmic problem (see Figure 1 for an example):

   For a line-graph G, find the length of the longest path that begins at v_1 and ends at v_n.

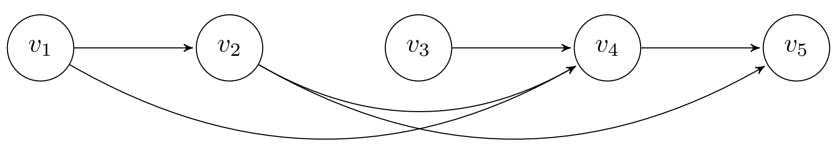

Figure 1: The correct answer for this line-graph is 3: the longest path from v_1 to v_n uses the three edges (v_1, v_2),(v_2, v_4), and (v_4, v_5).

(a) Show that the following algorithm does not correctly solve this problem, by giving an example of a line-graph on which it does not return the correct answer. In your example, say what the correct answer is and also what the algorithm finds.

Set $w = v_1$
Set $L = 0$
while there is an edge out of the node w **do**
    Choose the edge (w, v_j)
    for which j is as small as possible
    Set $w = v_j$
    Increase L by 1
end while
return L as the length of the longest path

(b) Give an algorithm that takes a line graph G and returns the length of the longest path that begins at v_1 and ends at v_n. The running time of your algorithm should be polynomial in n. Argue that your algorithm works correctly, and include a brief analysis of the running time.

4. Imagine you place a knight chess piece on a phone dial pad. This chess piece moves in an uppercase "L" shape: two steps horizontally followed by one vertically, or one step horizontally then two vertically:

Suppose you dial keys on the keypad using only hops a knight can make. Every time the knight lands on a key, we dial that key and make another hop. The starting position counts as being dialed. Give an efficient dynamic programming algorithm for the following problem: How many distinct numbers can you dial in N hops from a particular starting position? Specify the recursion equations on which your algorithm is based.