CSCI 3104 Assignment 6

10:00 - 10:50 Wanshan

Jackson Chen

March 14 2016

1. (a) LIS(a, 3, 2) = 1
   LIS(a, 5, infinity) = 3

   (b) LIS(a, 0, M) = 0
   LIS([], j, M) = 0

   (c)

$$LIS(a, j, M) = \begin{cases} 1 + LIS(a, j-1, a[j]) & \text{if } a[j] < M \\ LIS(a, j-1, M) & \text{if } a[j] >= M \end{cases}$$

   (d) A bottom-up scheme would consist of a table that solved LIS(a, j, M). The columns would consist of arrays [], a[0], a[0, 1], a[0, 1, 2], a[0, 1, 2, .., j-1]. All in all, there would be a total of j+1 columns. The rows would consist of the different values for M, from 0 to the maximum value in a + 1. The number of rows would be the max value in a + 2. The base cases would be 0. If the array to fill is an empty array, the longest increasing sequence is of length 0. If there is an array, but the numbers in the sequence cannot be any of the numbers in array (because they are bigger than M) then the longest sequence is also length 0. Each entry is filled out based on the upper and left values. Memo table:

	[]	**[1]**	**[1, 5]**	**[1, 5, 2]**	**[1, 5, 2, 3]**	**[1, 5, 2, 3, 8]**
0	0	0	0	0	0	0
1	0	1	1	1	1	1
2	0	1	1	1	1	1
3	0	1	1	2	2	2
4	0	1	1	2	3	3
5	0	1	1	2	3	3
6	0	1	2	2	3	3
7	0	1	2	2	3	3
8	0	1	2	2	3	3
9	0	1	2	2	3	4

2. There are a total of 21 paths from node 0 to node 13. In order to achieve this number, sum the number of paths it takes to get to the adjoining nodes. For example, nodes 0, 1, 2, 3, 4, 5 all have 1 possible path to reach them. To reach node 6, sum the two nodes that lead to it (0 and 1), so 1+1=2. Node 6 has 2 possible paths. Node 7 has 2+1+1=4. Node 8 4+1+1=6, and so on until node 13 is reached.