

Cs333 hw2 Report

1. Create a 2D array $n \times n$, n =lamps or outlets size, named checkMatchArr:
 - 1.1 Fill with 1 if the values are matchHS otherwise fill with 0
 - 1.2 Store the number of 1's, as int num;

2. Create 2D array numx2, to store i and j values of 1's in $n \times n$ array, named valuesOfIJ :
 - 2.1 Create a new 1D array with length num, named jValues
 - 2.2 Copy j values to jValues from valuesOfIJ
 - 2.3 Call longest increasing subsequence algorithm for jValues
 - 2.4 Print maxMatch and the sequence of that maxMatch.

3. Create longest increasing subsequence function, named LIS
 - 3.1 Take jValues array as input
 - 3.2 Create a `HashMap<ArrayList<Integer>, Integer>`, named resultHM
 - 3.3 Return result

Complexity is n^2 because we are iterating two for loops while filling the 2d array

EXAMPLE

Outlets, Lamps and n are as given:

Outlets = AA B4 BE DC 72
 Lamps = BE DC 72 AA B4

— int n = number of elements = 5

checkMatchArr and num are as given:

α CheckMatchArr: (5×5)

		AA 0	B4 1	BE 2	DC 3	72 4
BE 0	0	0	1	0	0	
DC 1	0	0	0	1	0	
72 2	0	0	0	0	1	
AA 3	1	0	0	0	0	
B4 4	0	1	0	0	0	

— int num = number of 1's = 5

ValuesOfIJ is as given:

α valuesOfIJ: (5×2)

		i 0	j 1
BE 0	0	2	
DC 1	1	3	
72 2	2	4	
AA 3	3	0	
B4 4	4	1	

jValues is as given:

α jValues: (5)

	BE 0	DC 1	72 2	AA 3	B4 4
2	3	4	0	1	

Create lamp