Dynamic Programming | Set 21 (Variations of LIS)

We have discussed Dynamic Programming solution for Longest Increasing Subsequence problem in this post and a O(nLogn) solution in this post. Following are commonly asked variations of the standard LIS problem.

1. Building Bridges: Consider a 2-D map with a horizontal river passing through its center. There are n cities on the southern bank with x-coordinates a(1) … a(n) and n cities on the northern bank with x-coordinates b(1) … b(n). You want to connect as many north-south pairs of cities as possible with bridges such that no two bridges cross. When connecting cities, you can only connect city i on the northern bank to city i on the southern bank.

```
8     1     4     3     5     2     6     7
<---- Cities on the other bank of river---->
-----------------------------------------
  <--------------- River--------------->
-----------------------------------------
1     2     3     4     5     6     7     8
<------- Cities on one bank of river------->
```

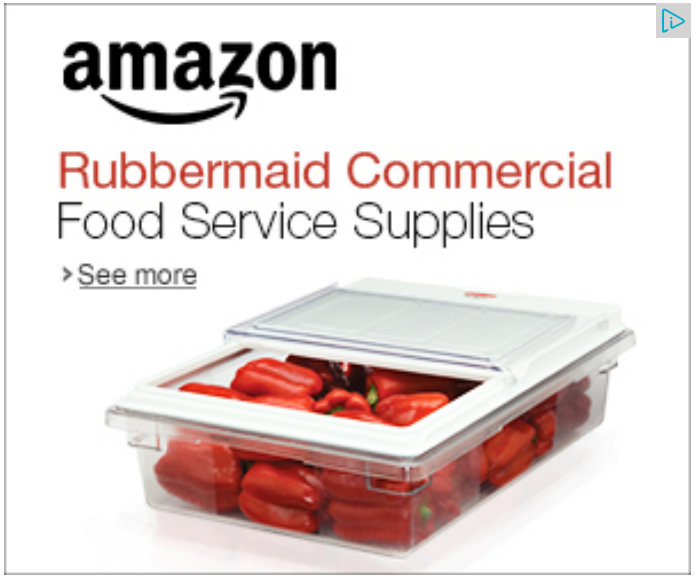
Source: Dynamic Programming Practice Problems. The link also has well explained solution for the problem.

2. Maximum Sum Increasing Subsequence: Given an array of n positive integers. Write a program to find the maximum sum subsequence of the given array such that the intgers in the subsequence are sorted in increasing order. For example, if input is {1, 101, 2, 3, 100, 4, 5}, then output should be {1, 2, 3, 100}. The solution to this problem has been published here.

3. The Longest Chain You are given pairs of numbers. In a pair, the first number is smaller with respect to the second number. Suppose you have two sets (a, b) and (c, d), the second set can follow the first set if b < c. So you can form a long chain in the similar fashion. Find the longest chain which can be formed. The solution to this problem has been published here.

4. Box Stacking You are given a set of n types of rectangular 3-D boxes, where the i^th box has height h(i), width w(i) and depth d(i) (all real numbers). You want to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base. It is also allowable to use multiple instances of the same type of box.
Source: Dynamic Programming Practice Problems. The link also has well explained solution for the problem.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Popular Posts

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion
- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

Writing code in comment? Please use **ideone.com** and share the link here.

7 Comments GeeksforGeeks

Sort by Newest

triveni • 9 days ago
i need to know program for num of nodes,connectivity,deletion of node.(the nu
for ex:the tree having 4 node after delet the num of node should be same)

xxmajia • 5 months ago
THANKS a lot

Mandeep • a year ago
[sourcecode language="java"]

```
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package DynaProg;

/**
*
* @author Pardeep
*/
public class buildingbridge {
public static void main(String ar[]){

int cr[]={8,1,4,3,5,2,6,7};
int oth[]={1,2,3,4,5,6,7,8};
int othpos[]=new int[cr.length];
for(int i=0;i<oth.length;i++) //find the occurence of oth[] element in cr[]
{
```

see more

GeeksforGeeks • 2 years ago

@Deep & @kalyan: The Box Stacking problem is also a variation of LIS proble maximum height stack.

1) We sort the boxes according to their increasing base areas (only a smaller another box).

2) After sorting the boxes, the problem is same as LIS.

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 37 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 41 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

H(i) = Maximum possible stack height with box i at top of stack
H(i) = { Max (H(j)) + height(i) } where j < i. If there is no such j then H(i) = heig
To get overall maximum height, we need to return max(H(i)) where 0 < i < n

We will soon publish it as a separate post.

kalyan • 2 years ago

can u expplain the box stacking problem pls ..

deep • 2 years ago

on the given link i am not getting The Box Stacking problem. can anyone expla

```
/* Paste your code here (You may delete these lines if not writing c
```

DEEP • 2 years ago

on the given link i am not getting The Box Stacking problem. can anyone expla