GeeksforGeeks

A computer science portal for geeks

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

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 ith 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.

Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: **Dynamic Programming**

Tweet ⟨ 1 ⟩   **g+1** ⟨ 1 ⟩

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

18 Comments   **GeeksforGeeks**   ① **Login** ⌄

♥ **Recommend**   ⤴ **Share**   **Sort by Newest** ⌄

Join the discussion…

typing.. · 4 months ago

What if we solve it by finding Longest Common Subsequence of the two arrays??

⌃ | ⌄ · Reply · Share ›

Gaurav pruthi · 5 months ago

Can we solve it using 'count inversions' ?

⌃ | ⌄ · Reply · Share ›

aa1992 · 7 months ago

what if in the bridges problem both the north and south pole have random coordinates .If north pole has {3,8,1,7,5,6,2,4} and south pole has {8,2,1,6,4,3,5,7}.how to find the intersection of the longest increasing sub sequence?

1 ⌃ | ⌄ · Reply · Share ›

> **subhtiwa** ➜ aa1992 · 7 months ago
>
> "If north pole has {3,8,1,7,5,6,2,4} and south pole has {8,2,1,6,4,3,5,7}" Here in north side you are using terminology that i will represent 3rd city as 1 or 4th city as 7 or 8th city as 4 and so on. But if you just re assign number starting from 1 onward, then your first array will be {1,2,3,...}.(you can do as it is in your hand).
>
> Find mapping of corresponding city in South side. now you will get 2nd array.
> Then find longest increasing sub sequence in 2nd array.
> If you didn't get the idea please again read my above statement and try to figure out your fault. I also faced same problem.
>
> ⌃ | ⌄ · Reply · Share ›

Anish · 9 months ago

I think the solution given to the Maximum Sum Increasing Subsequence problem only gives us the max sum inc sub-seq. The required output is an array which gives the max sum inc sub-seq.
Can anyone tell me the modification required to get the required output?

⌃ | ⌄ · Reply · Share ›

Karshit Jaiswal · 10 months ago

Solution to Building Bridges Problem :

http://ideone.com/Ps86ed

⌃ | ⌄ · Reply · Share ›

    amit → Karshit Jaiswal · 10 months ago

    kya bat hai

    ⌃ | ⌄ · Reply · Share ›

        Karshit Jaiswal → amit · 10 months ago

        khali.. :P

        ⌃ | ⌄ · Reply · Share ›

Suvodip Bhattacharya · a year ago

```
#include<stdio.h>
//#include<climits>
#include<iostream>
#include<climits>
using namespace std;
int max(int a,int b){ return ( a > b) ? a : b; }

int LISS(int arr[],int start,int current,int finish)
{

if( start >= finish ) return 0;

return ( current < arr[start] ) ? max( ( 1 + LISS(arr,start+1,arr[start],finish) ) ,
LISS(arr,start+1,current,finish) ) : LISS(arr,start+1,current,finish) ;

}

int main()
{
int arr[]={7,9,8,1,3,2,4,5,6};
int size=sizeof(arr)/sizeof(arr[0]);

cout<<liss(arr,0,int_min,size)<<endl; }="">
```

⌃ | ⌄ · Reply · Share ›

Guest · a year ago

```
#include<stdio.h>
//#include<climits>
#include<iostream>
#include<climits>
using namespace std;
```

```
using namespace std;
int max(int a,int b){ return ( a > b) ? a : b; }

int LISS(int arr[],int start,int current,int finish)
{
if( start >= finish ) return 0;
return ( current < arr[start] ) ? max( ( 1 +

LISS(arr,start+1,arr[start],finish) ) , LISS(arr,start+1,current,finish) ) :
LISS(arr,start+1,current,finish) ;

}

int main()
{
int arr[]={7,9,8,1,3,2,4,5,7,6};
int size=sizeof(arr)/sizeof(arr[0]);
cout<<liss(arr,0,int_min,size)<<endl; }="">
```

∧ | ∨ • Reply • Share ›

triveni · a year ago

i need to know program for num of nodes,connectivity,deletion of node.(the num of tree must be same

for ex:the tree having 4 node after delet the num of node should be same)

∧ | ∨ • Reply • Share ›

xxmajia · a year ago

THANKS a lot

∧ | ∨ • Reply • Share ›

Mandeep · 2 years 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};

5/5/2015   Dynamic Programming | Set 21 (Variations of LIS) - GeeksforGeeks

```
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

⌃ | ⌄ • Reply • Share ›

GeeksforGeeks · 3 years ago

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

1) We sort the boxes according to their increasing base areas (only a smaller base area box can be placed on top of another box).

2) After sorting the boxes, the problem is same as LIS.
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) = height(i)
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.

⌃ | ⌄ • Reply • Share ›

kalyan · 3 years ago

can u expplain the box stacking problem pls ..

⌃ | ⌄ • Reply • Share ›

deep · 3 years ago

on the given link i am not getting The Box Stacking problem. can anyone explain it?

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

⌃ | ⌄ • Reply • Share ›

DEEP · 3 years ago

on the given link i am not getting The Box Stacking problem. can anyone explain it?

⌃ | ⌄ • Reply • Share ›

mayuri → DEEP · 8 months ago

it is well explained here:
http://people.cs.clemson.edu/~...

⌃ | ⌄ • Reply • Share ›

✉ **Subscribe**   Ɗ **Add Disqus to your site**   ▷ **Privacy**

http://www.geeksforgeeks.org/dynamic-programming-set-14-variations-of-lis/   6/8

- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

- # Popular Posts

  - All permutations of a given string
  - Memory Layout of C Programs
  - Understanding "extern" keyword in C
  - Median of two sorted arrays
  - Tree traversal without recursion and without stack!
  - Structure Member Alignment, Padding and Data Packing
  - Intersection point of two Linked Lists
  - Lowest Common Ancestor in a BST.
  - Check if a binary tree is BST or not
  - Sorted Linked List to Balanced BST

- Follow @GeeksforGeeks

- # Recent Comments

  - lt_k

    i need help for coding this function in java...

[Java Programming Language](#) · [1 hour ago](#)

- [Piyush](#)

  What is the purpose of else if (recStack[*i])...

  [Detect Cycle in a Directed Graph](#) · [1 hour ago](#)

- [Andy Toh](#)

  My compile-time solution, which agrees with the...

  [Dynamic Programming | Set 16 (Floyd Warshall Algorithm)](#) · [1 hour ago](#)

- [lucy](#)

  because we first fill zero in first col and...

  [Dynamic Programming | Set 29 (Longest Common Substring)](#) · [2 hours ago](#)

- [lucy](#)

  @GeeksforGeeks i don't n know what is this long...

  [Dynamic Programming | Set 28 (Minimum insertions to form a palindrome)](#) · [3 hours ago](#)

- [manish](#)

  Because TAN is not a subsequence of RANT. ANT...

  [Given two strings, find if first string is a subsequence of second](#) · [3 hours ago](#)

-