Dynamic Programming | 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 a(i) on the northern bank to city b(i) on the southern bank. Maximum number of bridges that can be built to connect north-south pairs with the aforementioned constraints.

```
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------->
```

The values in the upper bank can be considered as the northern x-coordinates of the cities and the values in the bottom bank can be considered as the corresponding southern x-coordinates of the cities to which the northern x-coordinate city can be connected.

Examples:

```
Input : 6 4 2 1
        2 3 6 5
Output : Maximum number of bridges = 2
Explanation: Let the north-south x-coordinates
be written in increasing order.

1  2  3  4  5  6
  \  \
   \  \        For the north-south pairs
    \  \       (2, 6) and (1, 5)
     \  \      the bridges can be built.
      \  \     We can consider other pairs also,
       \  \    but then only one bridge can be built
        \  \   because more than one bridge built will
         \  \  then cross each other.
          \  \
1  2  3  4  5  6
```

```
Input : 8 1 4 3 5 2 6 7
        1 2 3 4 5 6 7 8
Output : Maximum number of bridges = 2
```

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

Approach: It is a variation of LIS problem. The following are the steps to solve the problem.

1. Sort the north-south pairs on the basis of increasing order of south x-coordinates.
2. If two south x-coordinates are same, then sort on the basis of increasing order of north x-coordinates.
3. Now find the Longest Increasing Subsequence of the north x-coordinates.
4. One thing to note that in the increasing subsequence a value can be greater as well as can be equal to its previous value.

We can also sort on the basis of north x-coordinates and find the LIS on the south x-coordinates.

```
// C++ implementation of building bridges
#include <bits/stdc++.h>

using namespace std;

// north-south coodinates
// of each City Pair
struct CityPairs
{
    int north, south;
};

// comparison function to sort
// the given set of CityPairs
bool compare(struct CityPairs a, struct CityPairs b)
{
    if (a.south == b.south)
        return a.north < b.north;
    return a.south < b.south;
}

// function to find the maximum number
// of bridges that can be built
int maxBridges(struct CityPairs values[], int n)
{
    int lis[n];
    for (int i=0; i<n; i++)
        lis[i] = 1;

    sort(values, values+n, compare);

    // logic of longest increasing subsequence
    // applied on the northern coordinates
    for (int i=1; i<n; i++)
        for (int j=0; j<i; j++)
            if (values[i].north >= values[j].north
                && lis[i] < 1 + lis[j])
                lis[i] = 1 + lis[j];


    int max = lis[0];
    for (int i=1; i<n; i++)
        if (max < lis[i])
            max = lis[i];

    // required number of bridges
    // that can be built
    return max;
}

// Driver program to test above
int main()
{
    struct CityPairs values[] = {{6, 2}, {4, 3}, {2, 6}, {1, 5}};
    int n = 4;
    cout << "Maximum number of bridges = "
```

```
            << maxBridges(values, n);
    return 0;
}
```

Run on IDE

Output:

```
Maximum number of bridges = 2
```

Time Complexity: $O(n^2)$

Problem References:
http://www.geeksforgeeks.org/dynamic-programming-set-14-variations-of-lis/

Solution References:
https://www.youtube.com/watch?v=w6tSmS86C4w

This article is contributed by **Ayush Jauhari**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

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

GATE CS Corner Company Wise Coding Practice

Dynamic Programming LIS

Recommended Posts:

Minimum insertions to sort an array
Dynamic Programming | Set 13 (Cutting a Rod)
Dynamic Programming | Set 22 (Box Stacking Problem)
Dynamic Programming | Set 3 (Longest Increasing Subsequence)
Dynamic Programming | Set 21 (Variations of LIS)

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!