Design and Analysis of Algorithms
Project Report

Aditya Bhagwat. **B00811694**

1. **Title of the Project :**

   Longest Increasing Subsequence (LIS) using Longest Common Subsequence(LCS) by using it on Engineering Applications.

2. **Problem Statement :**

   To perform the discovery of Longest Increasing Subsequence with the task of constructing maximum number of Bridges between banks of a river where the north and south coordinates are given and the goal is to maximize the Number (#) of Bridges to be built.

3. **Algorithm description :**

Algorithm for LIS using LCS :

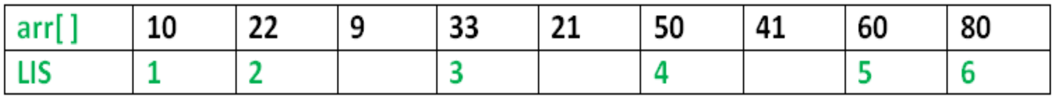
- Consider the array[] to be used and sort the array using any of the sorting algorithm like Heap Sort(or Merge Sort).
- Now place the sorted against the initial unsorted array by applying logic of Longest Common Subsequence on it.
- If the current value of Array and the new Array values turn out to be equal update the value in lcs[] array.
- Repeat the steps until all the numbers are checked.
- After above steps are completed check the maximum value in the lcs[] array and return this value as the maximum number of bridges.

Logic for LCS :

```
int lcs( char *X, char *Y, int m, int n )
{
if (m == 0 || n == 0)
return 0;
if (X[m-1] == Y[n-1])
return 1 + lcs(X, Y, m-1, n-1);
else
return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
}
```

Example :

Input : 10,22,9,33,21,50,41,60,80
Output : 5



arr[ ]	10	22	9	33	21	50	41	60	80
LIS	1	2		3		4		5	6

Design and Analysis of Algorithms
Project Report

Aditya Bhagwat. B00811694

Algorithm used for optimizing the number(#) of bridges to be built :

1. Consider there are ‘n’ cities on the North side of the bank and ‘n’ cities on the South side respectively.
2. Sort the north-south pairs on the basis of increasing order of south x-coordinates.
3. If two south x-coordinates are same, then sort on the basis of increasing order of north x-coordinates.
4. Now find the Longest Increasing Subsequence of the north x-coordinates.
5. One thing to note that in the increasing subsequence a value can be greater as well as can be equal to its previous value.

4. Results and discussion.

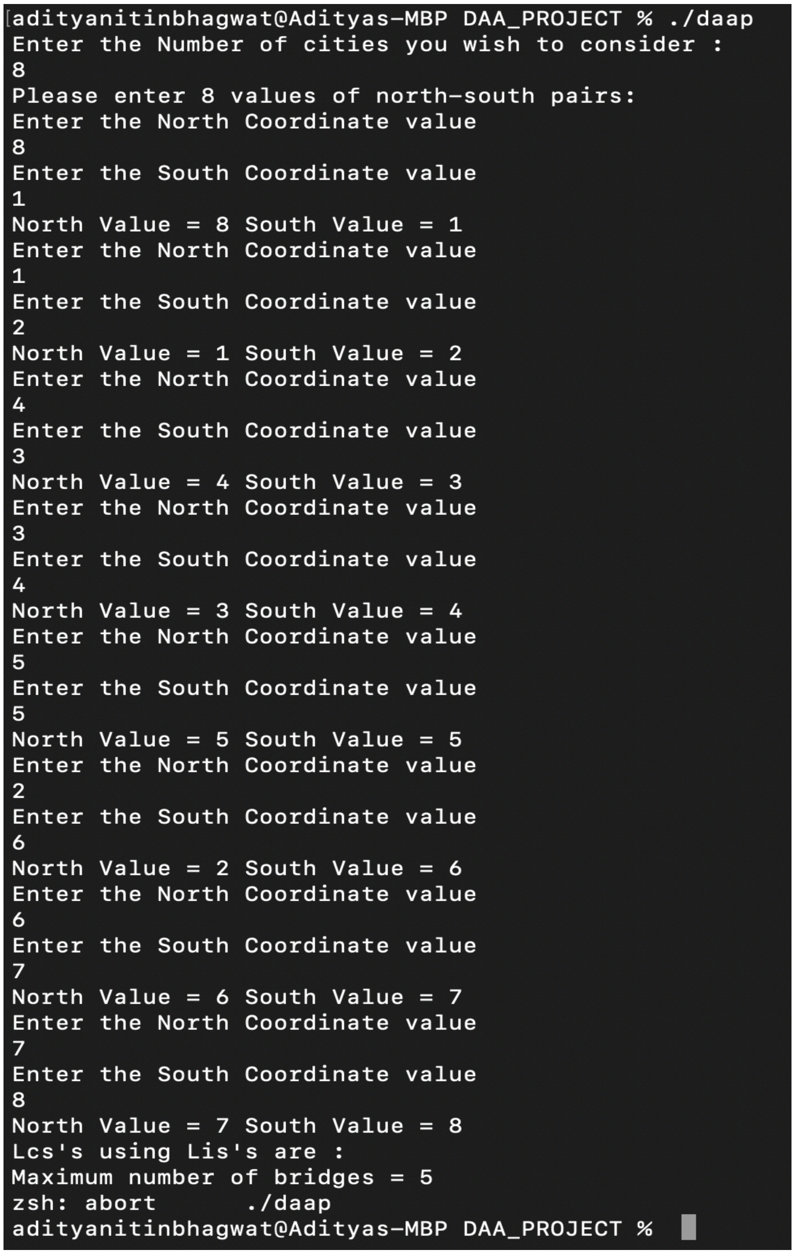
Implementation Status : Code working and implementing properly.
Implementation :

```
#include<iostream>
using namespace std;

// Define the Structure
struct Pairs
{
   //north and south coordinates of the city
         int north, south;
};

// returns max
bool compare(struct Pairs a, struct Pairs b)
{
         if (a.south == b.south)
                  return a.north < b.north;
         return a.south < b.south;
}

// to find the maximum bridges to be built
int maxBridges(struct Pairs values_pairs[], int n)
{
         int lcs[n];
         for (int i=0; i<n; i++)
                  lcs[i] = 1;

   // Logic of Longest Common Subsequence applied to the Numbers //

         sort(values_pairs, values_pairs+n, compare);
```

```
      for (int i=1; i<n; i++)
            for (int j=0; j<i; j++)
                  if (values_pairs[i].north >= values_pairs[j].north && lcs[i]
< 1 + lcs[j])
                        lcs[i] = 1 + lcs[j];


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


  cout<<"Lcs's using Lis's are :\n";




      return max;
  // returns the Maximum value of the # of Bridges
}

int main()
{
  int n = 0;
      struct Pairs values_pairs[] = {{6, 2}, {4, 3}, {2, 6}, {1, 5}};

  cout<<"Enter the Number of cities you wish to consider : \n";
  cin>>n;

  cout << "Please enter "<<n<<" values of north-south pairs: \n";

  for(int i=0;i<n;i++)
  {
    cout<<"Enter the North Coordinate value \n";
    cin >> values_pairs[i].north;
    cout<<"Enter the South Coordinate value \n";
    cin >> values_pairs[i].south;

    cout<<"North Value = "<< values_pairs[i].north<<"\t"<<"South Value =
"<<values_pairs[i].south<<"\n";
  }

  int new1 = maxBridges(values_pairs, n);
      cout << "Maximum number of bridges = "<<new1<<"\n";
```

```
        return 0;
}
```

Results :



```
[adityanitinbhagwat@Adityas-MBP DAA_PROJECT % ./daap
Enter the Number of cities you wish to consider :
8
Please enter 8 values of north-south pairs:
Enter the North Coordinate value
8
Enter the South Coordinate value
1
North Value = 8 South Value = 1
Enter the North Coordinate value
1
Enter the South Coordinate value
2
North Value = 1 South Value = 2
Enter the North Coordinate value
4
Enter the South Coordinate value
3
North Value = 4 South Value = 3
Enter the North Coordinate value
3
Enter the South Coordinate value
4
North Value = 3 South Value = 4
Enter the North Coordinate value
5
Enter the South Coordinate value
5
North Value = 5 South Value = 5
Enter the North Coordinate value
2
Enter the South Coordinate value
6
North Value = 2 South Value = 6
Enter the North Coordinate value
6
Enter the South Coordinate value
7
North Value = 6 South Value = 7
Enter the North Coordinate value
7
Enter the South Coordinate value
8
North Value = 7 South Value = 8
Lcs's using Lis's are :
Maximum number of bridges = 5
zsh: abort      ./daap
adityanitinbhagwat@Adityas-MBP DAA_PROJECT %
```

Time Complexity : O(n^2)

5. References :

- https://www.geeksforgeeks.org/dynamic-programming-building-bridges/
- https://stackoverflow.com/questions/7288585/building-bridges-problem-how-to-apply-longest-increasing-subsequence
- https://www.sanfoundry.com/dynamic-programming-solutions-building-bridges-problem/