

## Dynamic Programming | Set 3 (Longest Increasing Subsequence)

We have discussed Overlapping Subproblems and Optimal Substructure properties in [Set 1](#) and [Set 2](#) respectively.

Let us discuss Longest Increasing Subsequence (LIS) problem as an example problem that can be solved using Dynamic Programming.

The longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order. For example, length of LIS for { 10, 22, 9, 33, 21, 50, 41, 60, 80 } is 6 and LIS is {10, 22, 33, 50, 60, 80}.

**We strongly recommend that you click here and**

[Start Coding Today](#)

Home Equity  
Line of Credit

HOW MUCH DO YOU NEED?

\$10,000

\$15,000

\$20,000

## practice it, before moving on to the solution.

### Optimal Substructure:

Let  $arr[0..n-1]$  be the input array and  $L(i)$  be the length of the LIS till index  $i$  such that  $arr[i]$  is part of LIS and  $arr[i]$  is the last element in LIS, then  $L(i)$  can be recursively written as.

$L(i) = \{ 1 + \text{Max} ( L(j) ) \}$  where  $j < i$  and  $arr[j] < arr[i]$  and if there is no such  $j$  then  $L(i) = 1$

To get LIS of a given array, we need to return  $\max(L(i))$  where  $0 < i < n$ . So the LIS problem has optimal substructure property as the main problem can be solved using solutions to subproblems.

### Overlapping Subproblems:

Following is simple recursive implementation of the LIS problem. The

lis ending with every element is returned using `max_ending_here`. The overall lis is returned using pointer to a variable `max`.

C/C++

Java

Python

```
/* A Naive C/C++ recursive implementation of LIS problem
#include<stdio.h>
#include<stdlib.h>

/* To make use of recursive calls, this function must return
two things:
```

\$25,000	\$30,000	\$35,000
\$40,000	\$45,000	\$50,000
\$55,000	\$60,000	\$65,000
\$70,000	\$75,000	\$80,000

Calculate Payment >

**lendingtree**

Terms & Conditions apply. NMLS#1138

### Popular Posts

- Top 10 Algorithms and Data Structures for Competitive Programming
- Top 10 algorithms in Interview Questions
- How to begin with Competitive

```

1) Length of LIS ending with element arr[n-1]. We use
   max_ending_here for this purpose
2) Overall maximum as the LIS may end with an element
   before arr[n-1] max_ref is used this purpose.
The value of LIS of full array of size n is stored in
*max_ref which is our final result */
int _lis( int arr[], int n, int *max_ref)
{
    /* Base case */
    if (n == 1)
        return 1;

    // 'max_ending_here' is length of LIS ending with arr[n-1]
    int res, max_ending_here = 1;

    /* Recursively get all LIS ending with arr[0], arr[1]
       arr[n-2]. If arr[i-1] is smaller than arr[n-1],
       max ending with arr[n-1] needs to be updated, then
       update it */
    for (int i = 1; i < n; i++)
    {
        res = _lis(arr, i, max_ref);
        if (arr[i-1] < arr[n-1] && res + 1 > max_ending_here)
            max_ending_here = res + 1;
    }

    // Compare max_ending_here with the overall max. And
    // update the overall max if needed
    if (*max_ref < max_ending_here)
        *max_ref = max_ending_here;

    // Return length of LIS ending with arr[n-1]
    return max_ending_here;
}

// The wrapper function for _lis()
int lis(int arr[], int n)
{
    // The max variable holds the result
    int max = 1;

```

## How to begin with Competitive Programming?

- Step by Step Guide for Placement Preparation
- Reflection in Java
- Memory Layout of C Programs
- Heavy Light Decomposition
- Sorted Linked List to Balanced BST
- Generics in Java
- Aho-Corasick Algorithm for Pattern Searching
- Insertion Sort , Binary Search , QuickSort , MergeSort , HeapSort

Track Corporate Web  
Activity with GFI  
WebMonitor® . Free T

gfi.com

```

// The function _lis() stores its result in max
_lis( arr, n, &max );

// returns max
return max;
}

/* Driver program to test above function */
int main()
{
    int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Length of lis is %d\n",
           lis( arr, n ));
    return 0;
}

```

Run on IDE

Length of lis is 5

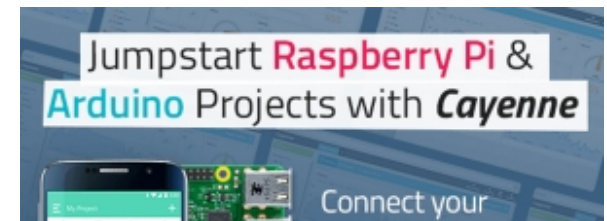
Considering the above implementation, following is recursion tree for an array of size 4. lis(n) gives us the length of LIS for arr[].

```

          lis(4)
        /   |   \
      lis(3) lis(2) lis(1)
     /  \   /
  lis(2) lis(1) lis(1)
 /
lis(1)

```

- Common Interview Puzzles
- Interview Experiences
- Advanced Data Structures
- Design Patterns
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Geometric Algorithms
- Searching
- Sorting
- Hashing
- Analysis of Algorithms
- Mathematical Algorithms
- Randomized Algorithms
- Recursion



We can see that there are many subproblems which are solved again and again. So this problem has Overlapping Substructure property and recomputation of same subproblems can be avoided by either using Memoization or Tabulation. Following is a tabulated implementation for the LIS problem.

C/C++

Java

Python

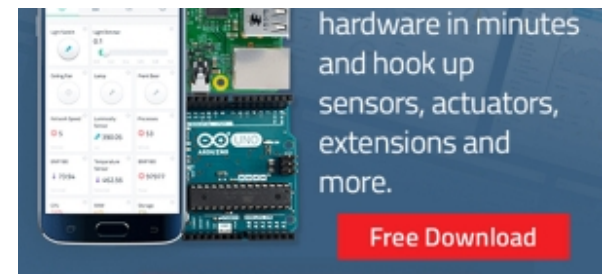
```
/* Dynamic Programming C/C++ implementation of LIS problem */
#include<stdio.h>
#include<stdlib.h>

/* lis() returns the length of the longest increasing
subsequence in arr[] of size n */
int lis( int arr[], int n )
{
    int *lis, i, j, max = 0;
    lis = (int*) malloc ( sizeof( int ) * n );

    /* Initialize LIS values for all indexes */
    for (i = 0; i < n; i++ )
        lis[i] = 1;

    /* Compute optimized LIS values in bottom up manner */
    for (i = 1; i < n; i++ )
        for (j = 0; j < i; j++ )
            if ( arr[i] > arr[j] && lis[i] < lis[j] + 1)
                lis[i] = lis[j] + 1;

    /* Pick maximum of all LIS values */
    for (i = 0; i < n; i++ )
        if (max < lis[i])
            max = lis[i];
}
```



## Tags

Accolite Adobe Advance Data Structures Advanced Data Structures Amazon array D-E-Shaw Directi Divide and Conquer Dynamic Programming Flipkart GATE GATE-CS-DS-&-Algo GATE-CS-Older GFacts Goldman Sachs Google Graph Greedy Algorithm Hashing Interview Experience Java MakeMyTrip MAQ Software MathematicalAlgo Matrix Microsoft Morgan Stanley Operating systems Oracle Pattern Searching Recursion Samsung SAP Labs SnapDeal stack STL Zoho

Subscribe and Never Miss an Article

```

/* Free memory to avoid memory leak */
free(lis);

return max;
}

/* Driver program to test above function */
int main()
{
    int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Length of lis is %d\n", lis( arr, n ) );
    return 0;
}

```

Run on IDE

Output:

Length of lis is 5

Note that the time complexity of the above Dynamic Programming (DP) solution is  $O(n^2)$  and there is a  $O(n \log n)$  solution for the LIS problem. We have not discussed the  $O(n \log n)$  solution here as the purpose of this post is to explain Dynamic Programming with a simple example. See below post for  $O(n \log n)$  solution.

### Longest Increasing Subsequence Size ( $N \log N$ )

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

Email Address

Subscribe

Recent Comments

Like us on Facebook

Follow us on Twitter

Subscribe on YouTube

### All Categories

Select Category

## Company Wise Coding Practice Coding Practice

## Topic Wise

357 Comments Category: Dynamic Programming Tags: Dynamic Programming

---

### Related Posts:

---

- All ways to add parenthesis for evaluation
- Count all increasing subsequences
- Count distinct occurrences as a subsequence
- Longest repeating and non-overlapping substring

- Find minimum sum such that one of every three consecutive elements is taken
- Count Distinct Subsequences
- Non-crossing lines to connect points in a circle
- Count digit groupings of a number with given constraints

Previous post in category

Next post in category

(Login to Rate and Mark)

3

Average Difficulty : **3/5.0**  
Based on **176** vote(s)



Add to TODO List



Mark as DONE

Writing code in comment? Please use [code.geeksforgeeks.org](https://code.geeksforgeeks.org), generate link and share the link here.

**357 Comments**

**GeeksforGeeks**

♥ Recommend **11**  Share

Join the discussion...



**TheFlash** • 3 days ago





```
public static void main(String[] args) {  
    // TODO code application logic here  
    int arr[] = {10, 22, 9, 33, 5, 4};  
    int length = 1;  
    int curr_lenght, last_high ;  
    for (int i = 0; i < arr.length; i++){  
        curr_lenght = 1;  
        last_high = arr[i];  
        for (int j = i+1; j<arr.length;j++){ if(arr[j]>""> last_high){  
            last_high = arr[j];  
            curr_lenght++;  
        }  
        if (curr_lenght > length)  
            length = curr_lenght;  
    }  
}  
  
System.out.println("length = " + length);  
}
```

Is this solution correct? How DP solution is better than this?

^ | v • Reply • Share ›



**ds.algos143** • 18 days ago

DP solution using recursion

```
//an intermmmediate array to store the results  
private static int[] result = new int[8];
```

```
public static int findInON2Recur(int[] arr, int n) {  
  
    // base case  
    if (n == 1) {  
        return 1;  
    } else if(result[n-1] != -1) {  
        return result[n-1];  
    }  
  
    .....  
    .....  
    //after the loop and before updating the max_ref  
    // Compare max_ending_here with the overall max. And  
    // update the overall max if needed  
    result[n-1] = max_ending_here;  
    ^ | v • Reply • Share ›
```



**Aza Tulebergenov** • 18 days ago

Why in this check "if arr[i] > arr[j] and lis[i] < lis[j] + 1 :", we have lis

^ | v • Reply • Share ›



**ds.algos143** ➔ Aza Tulebergenov • 18 days ago

1.  $lis[i] < lis[j] + 1$ , to update only if the new value will give tl

2. if if arr[i] > arr[j] and  $lis[i] < lis[j] + 1$  condition is true, you  
the lis found till arr[0] to arr[i-1]

^ | v • Reply • Share ›



**Mudit Sharma** • 25 days ago

Why this for loop ?

```
for (i = 0; i < n; i++ )
```

```
if (max < lis[i])
```

```
max = lis[i];
```

Max will always be at lis[n]

^ | v • Reply • Share ›



**Sadiq Husain Khan** ➔ Mudit Sharma • 5 days ago

not in all cases e.g. 10, 22, 9, 33, 5, 4

^ | v • Reply • Share ›



**st** • a month ago

```
int max=0,n=arr.length,count=0;
```

```
int k=0,j=0,i=0;
```

```
for(i=0;i<n;i++) {="" count="1;" k="i;" j="i;" while(j<n)="" {="" if(arr[j]
```

```
{
```

```
k=j;
```

```
j++;
```

```
count++;
```

```
}
```

```
else
```

```
{
```

```
j++;
```

```
}
```

```
}
```

```
,  
if(count>max) max=count;  
}  
System.out.println(count);
```

^ | v • Reply • Share ›



**st** → st • a month ago

\*correction : System.out.println(max);

^ | v • Reply • Share ›



**Aakanksha Sahu** • a month ago

<http://code.geeksforgeeks.org/...>

correct me if I am wrong.

^ | v • Reply • Share ›



**Ayushrazz Choudhary** • a month ago

Solution with no extra space

<http://ideone.com/ofKIY6>

^ | v • Reply • Share ›



**malik** • a month ago

We can use same temp arr for printing,

```
for(int i=n-1;i>=0;i--){  
if(temp[i]==max)sout(arr[i]);  
max--;  
}
```

Hope it Helps :)

^ | v • Reply • Share ›



**Ashish Jaiswal** • 2 months ago

printing order also: <http://codepad.org/8pRK24WW>

^ | v • Reply • Share ›



**Avalokita** • 2 months ago

what about this code for naive approach:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int lis(int *a, int n)
```

```
{
```

```
int i=0,ctr=1,temp=a[0];
```

```
for(i=0;i<n-1;i++) {="" if(temp<a[i+1]){="" ctr++;="" temp="a[i+1];}"
```

```
to="" test="" above="" function="" *="" int="" main()="" {="" int="" a
```

```
};="" int="" n="sizeof(arr)/sizeof(arr[0]);" printf("length="" of="" lis='
```

```
0;="" }="">
```

^ | v • Reply • Share ›



**nitin** • 3 months ago

```
public static int getSequenceAtIndex(int[] arr) {
```

```
int[] result = new int[arr.length];
```

```
Arrays.fill(result, 1);
```

```
int max = 1;
```

```
for (int i = 1; i < arr.length; i++) {
```

```
for (int j = 0; j < i; j++) {
```

```

if (arr[j] < arr[i]) {
result[i] = Math.max(result[i], result[j] + 1);
if(result[i] > max){
max = result[i];
}
}
}
}
return max;
}

```

^ | v • Reply • Share ›



**Siva Subramanian** • 3 months ago

Here is the solution in java .

```

public class LongIncreaseSeq {
static int[] lis(int arr[],int n)
{
int[] parent = new int[n];
for(int k=0;k<parent.length;k++) {="" parent[k]="-1;" }="" int[]="" lis=
lis[0]="0;" int="" ceil="0;" for(int="" i="1;i<arr.length;i++)" {="" if(a
parent[i]="lis[ceil-1];" }="" else="" {="" int="" newceil="findceil(lis,i,a
parent[i]="lis[newceil-1];" }="" }="" int="" max="0;" for(int="" i="0;i&
max="lis[i];" }="" }="" int="" m="max;" while(m="">=0)
{
int arrpointer = arr[m];
result[m]=arrpointer;

```

```
m=parent[m];  
max=m;  
}
```

~~return result;~~

[see more](#)

^ | v • Reply • Share ›



**Jayesh Tambe** • 3 months ago

I tried ur solution. I needs slight update.

In 3rd for loop, where u start j = 0 , i think it needs to start from j =

^ | v • Reply • Share ›



**siva** • 3 months ago

O(n) approach. without using DP or recurssion

<http://code.geeksforgeeks.org/...>

^ | v • Reply • Share ›



**gajam** → siva • 3 months ago

It is wrong bro... try for this array -> 1,10,2,3,4,5,6,7,8

1 ^ | v • Reply • Share ›



**Sunny Arora** → gajam • a month ago

it is correct !. why it is wrong?

^ | v • Reply • Share ›



**LALIT** → Sunny Arora • a month ago

try 10 1 11 2 3 4 5  
answer shud be 5  
but its 2 in the above code

^ | v • Reply • Share ›



**Bhavyaa Bansal** • 3 months ago

What is the complexity of this solution for LIS?

<http://code.geeksforgeeks.org/...>

^ | v • Reply • Share ›



**evolver** • 4 months ago

C++ non recursive  $O(n^2)$  approach :

<http://code.geeksforgeeks.org/...>

^ | v • Reply • Share ›



**Ankit Mishra** • 4 months ago

No Need to write such complex logic and third parameter in Recu  
use this =====>

```
public int longestIncreasingSub(int[] arr,int i){  
    int maxAtThisLevel=-1;  
    if(i==0) return 1;  
    for(int j=i-1;j>=0;j--){  
        int valueAtNode=1;  
        if(arr[i]>arr[j]) valueAtNode=longestIncreasingSub(arr,j)+1;  
        if(valueAtNode>maxAtThisLevel) maxAtThisLevel=valueAtNode;  
    }
```



```
if(valueAtNode>maxAtThisLevel) maxAtThisLevel=valueAtNode;  
}  
return maxAtThisLevel;  
}
```

```
public static void main(String[] args) {  
  
int[] arr={ 10, 22, 9, 33, 21, 50, 41, 60};  
Lis lis = new Lis();  
System.out.println(lis.longestIncreasingSub(arr, arr.length-1));  
  
}
```

^ | v • Reply • Share ›



**Karan Kapoor** • 4 months ago

Top Down : <http://ideone.com/zkXxae>

^ | v • Reply • Share ›



**midnight\_stalker** → Karan Kapoor • 3 months ago

how to change a bottom-up approach into top-down.recurs

^ | v • Reply • Share ›



**Karan Kapoor** • 4 months ago

<http://ideone.com/pzhPZC>

^ | v • Reply • Share ›



**Princess** • 4 months ago

Please explain this !



Please explain this :

^ | v • Reply • Share ›



**Praveen Kumar Mummidivarapu** → Princess • 4 months ago

what?

^ | v • Reply • Share ›



**hardcoder!** • 4 months ago

it can be solved in  $n \log n$  using binary indexed tree..  $dp[i] = \max \text{len}$  index  $i$  we are checking all the  $j$  from 0 to  $i-1$  and updating answer query and update in BIT.. as we only update the answer for values stored so far in  $a[j]$  which is less than  $a[i]$  and then update it.. seg

^ | v • Reply • Share ›



**Abhinav srivastava** • 4 months ago

someone can please explain, how and in what condition can we lc

^ | v • Reply • Share ›



**.NetGeek** • 5 months ago

@GeeksForGeeks: Could you please elaborate on the recursive l  
?

Thanks :)

1 ^ | v • Reply • Share ›



**Praveen Kumar Mummidivarapu** → .NetGeek • 4 months ago

<http://code.geeksforgeeks.org/...>

^ | v • Reply • Share ›



**Abhishek Kumar** → Praveen Kumar Mummidivarapu • 4 r

your code doesn't work if the first element is a very test case :-

15

1000000000 22 9 33 21 50 41 60 80 90 90 100 103

your output - 1

correct output - 9

^ | v • Reply • Share ›



**ankit agarwal** • 5 months ago

`arr[i-1]<arr[n-1],i could't="" understand="" the="" logic="" behind=""`

^ | v • Reply • Share ›



**rexin** 冲 • 5 months ago

Here the code for printing of the numbers also

<https://ideone.com/ivlAuc>

2 ^ | v • Reply • Share ›



**max** • 5 months ago

what is the need for `lis[i]<lis[j]+1 in="" dp="" solution.="">`

^ | v • Reply • Share ›



**rexin** 冲 → max • 5 months ago

To maintain the number of node that is less than or equal to

10 maintain the number of node that is less than or equal to

1 ^ | v • Reply • Share ›



**max** → rexin冲 • 5 months ago

can u give an example where it is needed and why

^ | v • Reply • Share ›



**rexin**冲 → max • 5 months ago

Ok if u not want to write the condition then u  
 $lis[i] = \max(lis[i], lis[j] + 1)$

1 ^ | v • Reply • Share ›



**Binary** • 5 months ago

I think it makes sense to make recursive calls only if the value is less than the current value. This optimization in recursive version would be,

```
for (int i = 1; i < n; i++)  
{  
    // Go recursive only if the below condition is satisfied  
    if (arr[i-1] < arr[n-1] )  
    {  
        res = _lis(arr, i, max_ref);  
        if( res + 1 > max_ending_here)  
            max_ending_here = res + 1;  
    }  
}
```

^ | v • Reply • Share ›



**raja** • 5 months ago

easy code:

```
#include<bits/stdc++.h>

using namespace std;

int lis(int a[],int n)

{

int dp[n];int i,j;

for (i = 0; i < n; i++ )

dp[i] = 1;

for(i=1;i<n;i++) {="" for(j="0;j<i;j++)" {="" if(a[i]>a[j])

dp[i]=max(dp[i],dp[j]+1);

}
```

[see more](#)

^ | v • Reply • Share ›



**Narender Soorineeda** • 5 months ago

//java solution

```
public static int lis(int[] arr, int p){
```

```
if(p == 0){
```

```

if(p == 0){
return 1;
}
int max = 1;
for(int i = 0;i<p;i++){ if(arr[i]<= arr[p]){ int temp=1+lis(a
max = temp;
}
}
return max;
}

```

^ | v • Reply • Share ›



**.NetGeek** • 5 months ago

C# Implementation: <http://ideone.com/5DqREw>

1 ^ | v • Reply • Share ›



**rahul kumar** • 5 months ago

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n,a[100],b[100],t=0;
```

```
cin>>n;
```

```

for(int i=0;i<n;i++) cin="">a[i];

for(int i=0;i<n;i++) b[i]="1;" for(int="" i="1;i<n;i++)" {"="" for(int="" j
b[i]="max(b[i],b[j]+1);" if(b[i]==">t)

t=b[i];

}

}

cout<<"The longest increasing subsequence is: ";

cout<<t<<endl; }="">

```

^ | v • Reply • Share ›



**Leo Fernandez** • 5 months ago

A solution in c# that prints the list and uses a dictionary for memo  
Please let me know if you see any errors.

^ | v • Reply • Share ›



**Yuan Sun** • 5 months ago

I tried with int arr[] = { 10, 9, 8, 7, 9, 5, 11, 0 } with recursive algorit

^ | v • Reply • Share ›



**Angel** ➔ Yuan Sun • 5 months ago

3 is correct..(8,9,11 or 7,9,11) are the longest increasing s

^ | v • Reply • Share ›



**angelleecash** • 5 months ago

```
def longest_increasing_sub_sequence(a, offset=0, prev=-1, c=0):  
    if offset >= len(a):  
  
        return c  
  
        take_it = c  
  
        if prev < 0:  
  
            take_it = 1  
  
        else:  
  
            if a[offset] > a[prev]:  
  
                take_it += 1  
  
            take_it = longest_increasing_sub_sequence(a, offset+1, offset, ta  
            leave_it = longest_increasing_sub_sequence(a, offset+1, prev, c)  
  
            return max(take_it, leave_it)  
  
    print longest_increasing_sub_sequence([10, 22, 9, 33, 21, 50, 41,  
    ^ | v • Reply • Share ›
```



**The Nitian** • 5 months ago





If the sequence is 6 15 29 80 8 then memorization will not work. T

^ | v • Reply • Share ›



**Deepak Mundhada** • 6 months ago

I tried to print LI sequence also along with length and below is the  
eg 1)

For Input - 10 22 33 51 21 25 39 42 43 44 54 65 76 87

LIS as per code is - 10 22 33 51 42 43 44 54 65 76 87

Instead of - 10 22 33 39 42 43 44 54 65 76 87

eg 2)

For Input - 0 8 4 12 2 10 6 14 1 9 5 13 3 11 7 15

LIS as per code is - 0 8 12 14 13 15

Instead of - 0, 2, 6, 9, 11, 15 or 0, 4, 6, 9, 11, 15 or 0, 4, 6, 9, 13, 1

Any thought anybody?

^ | v • Reply • Share ›

Load more comments



Subscribe



Add Disqus to your site

Privacy



@geeksforgeeks, Some rights reserved

Contact Us!

About Us!

Advertise with us!