

Longest Increasing Subsequence: Source Code

Name : K V Sagar

- The aim is to find the longest increasing subsequence (LIS) of an unsorted array.
- To do this we first create a dummy/temp array of the same size as the actual array.
- Assign 1s to all indices of the dummy array.
- Then traverse along the actual array checking for the LIS.
- The below mentioned condition is used to search for the LIS

```
for (i = 1; i < n; i++)  
    for (j = 0; j < i; j++)  
        if (a[i] > a[j] && b[i] < b[j] + 1)  
            b[i] = b[j] + 1;
```

- Consider an array $a[5] = \{2, -2, 3, 1, 5\}$.
- Here, the dummy array will be $b[5] = \{1, 1, 1, 1, 1\}$.
- The following will show the multiple iterations of the snippet,
 1. $a[0] = 2 < a[1] = -2$ since the condition is not satisfied $b[] = \{1, 1, 1, 1, 1\}$.
 2. $a[0] = 2 < a[2] = 3$ since condition is satisfied $b[] = \{1, 1, 2, 1, 1\}$.
 3. $a[0] = 2 < a[3] = 1$ since condition is not satisfied $b[] = \{1, 1, 2, 1, 1\}$.
 4. $a[0] = 2 < a[4] = 5$ since condition is satisfied $b[] = \{1, 1, 2, 1, 3\}$.
- The dummy array is now, $b[] = \{1, 1, 2, 1, 3\}$. The max value stored in this array is the size of the LIS. Here, it is 3.