```java
public class Longest_Increasing_Subsequence {
   static int max_ref; // overall maximum LIS

   static int _lis(int arr[], int n) {
       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], and
           max ending with arr[n-1] needs to be updated, then
           update it */
        for (int i = 1; i < n; i++) {
            res = _lis(arr, i);
            if (arr[i-1] < arr[n-1] && res + 1 > max_ending_here)
                max_ending_here = res + 1;
        }

        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()
    static int lis(int arr[], int n) {
         max_ref = 1;
        _lis( arr, n);
        return max_ref;
    }

    // driver program to test above functions
    public static void main(String args[]) {
        int arr[] = { 6,5, 10, 22, 9, 33, 21, 50, 41, 60 };
        int n = arr.length;
        System.out.println("Lis's length: " + lis(arr, n));
    }
 }
```