Maximum length of longest increasing sub sequence

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

int LongestIncreasingSubsequence(int *arr, int size)
{
        int *lis, i, j, maxLen = 0;
        lis = (int *)malloc(sizeof(int) * size);

        for(i = 0; i < size; i++)
                lis[i] = 1;

        for (i = 1; i < size; i++ )
      for (j = 0; j < i; j++ )
        if ( arr[i] > arr[j] && lis[i] < lis[j] + 1)
           lis[i] = lis[j] + 1;

   for (i = 0; i < size; i++ )
      if (maxLen < lis[i])
         maxLen = lis[i];
   free(lis);
   return maxLen;
}

int main()
{
        int *arr, size;
        printf("Enter size of the array\n");
        scanf("%d", &size);

        //allocate memory
        arr = (int *)malloc(sizeof(int) * size);

        printf("Enter elements in array\n");
        for(int index = 0; index < size; index++)
                scanf("%d", &arr[index]);

        printf("Length of longest increasing sub sequence is = %d",
                LongestIncreasingSubsequence(arr, size));

        return 0;
}
```

Time complexity: $O(n^2)$
Space complexity: O(n)