Question 3

Date:

//using arrays
// using binary search

if $X[i]$ > lastelement in S,
    add $X[i]$ into S
else
    find smallest element ~~in~~ larger than $X[i]$, $S[k] < X[$
    and $X[i] \le S[k+1]$, replace $S[k+1]$ with $X[i]$
// The length of S is length of LIS of X

LIS$(X, n)$
  $L = 0$;
  for $i = 1, 2, \dots, n$
  BinSearch for largest positive $j \le L$ s.t $X[M[j]] < X[i$
     $j = 0$ if no value exists
  $P[i] = M[j]$
  if $j == L$ or || $X[i] < X[M[j+1]]$
     $M[j+1] = i$
     $L = \max(L, j+1)$

// $M[j]$ stores the position ~~of~~ $k$ of smallest value $X[k]$ such that there is an increasing subsequence of length $j$ that ends @ $X[k]$ when $k \le i$ $(j \le k \le i)$

// $P[k]$ will store the position of the predecessor of $X[k]$ in longest increasing subsequence that ends @ $X[k]$

// $L$ is length of LIS. Since the algorithm performs only one binary search, the total time will be $O(n \log n)$