EMP No: 2577108
Name: Pramod

Algorithm of Longest Increasing Subsequence:

The algorithm for finding the longest increasing subsequence in an array can be described
as follows:
1.Initialize an empty list called piles to store the piles of cards.

2.Iterate over each number ‘num’ in the input array ‘nums’:
- Create a new Pile object with num as the top element.
- Use binary search to find the index where the new pile should be inserted in the piles

list:
➢ Initialize left as 0 and right as the size of piles minus 1.
➢ While left is less than or equal to right, do:
➢ Calculate the middle index as mid = left + (right - left) / 2.
➢ If piles[mid].top is less than num, update left = mid + 1.
➢ Otherwise, update right = mid - 1.
➢ After the binary search, the correct position to insert the new pile is left.
➢ If left is equal to the size of piles, add the new pile to the end of the list.
➢ Otherwise, update the existing pile at index left with the new pile.

3.Create a new list called ‘lis’.

4.Iterate over each pile in the piles list:
➢ Add the top element of each pile to the ‘lis’ list.

5.Return the ‘lis’ list, which represents the longest increasing subsequence in the input array
‘nums’.