[Longest Increasing Subsequence using Binary Serach(lower-bound)](https://leetcode.com/problems/longest-increasing-subsequence/)

**lower\_bound:** lower\_bound(nums.begin(), nums.end(), arr[i])- nums.begin();

-> It returns index of arr[i] if it is present in nums else returns the index of element greater than arr[i].

**Approach:** If we make a vector using lower\_bound on all the elements of the given array, then vector that we get in the last will have size same as LIS.

1. Make a vector LB and store nums[0]; nums is the array for which we want to find LIS.
2. Iterate over given array, and if element is greater than the LB.back() then directly push it else find the index using lower\_bound and then insert at that index.

**Example**: [0, 1, 0, 3, 2, 3]

LB=> [0] -> [0, 1] -> [0, 1, 3] -> [0, 1, 2] -> [0, 1, 2, 3]

o/p: 4