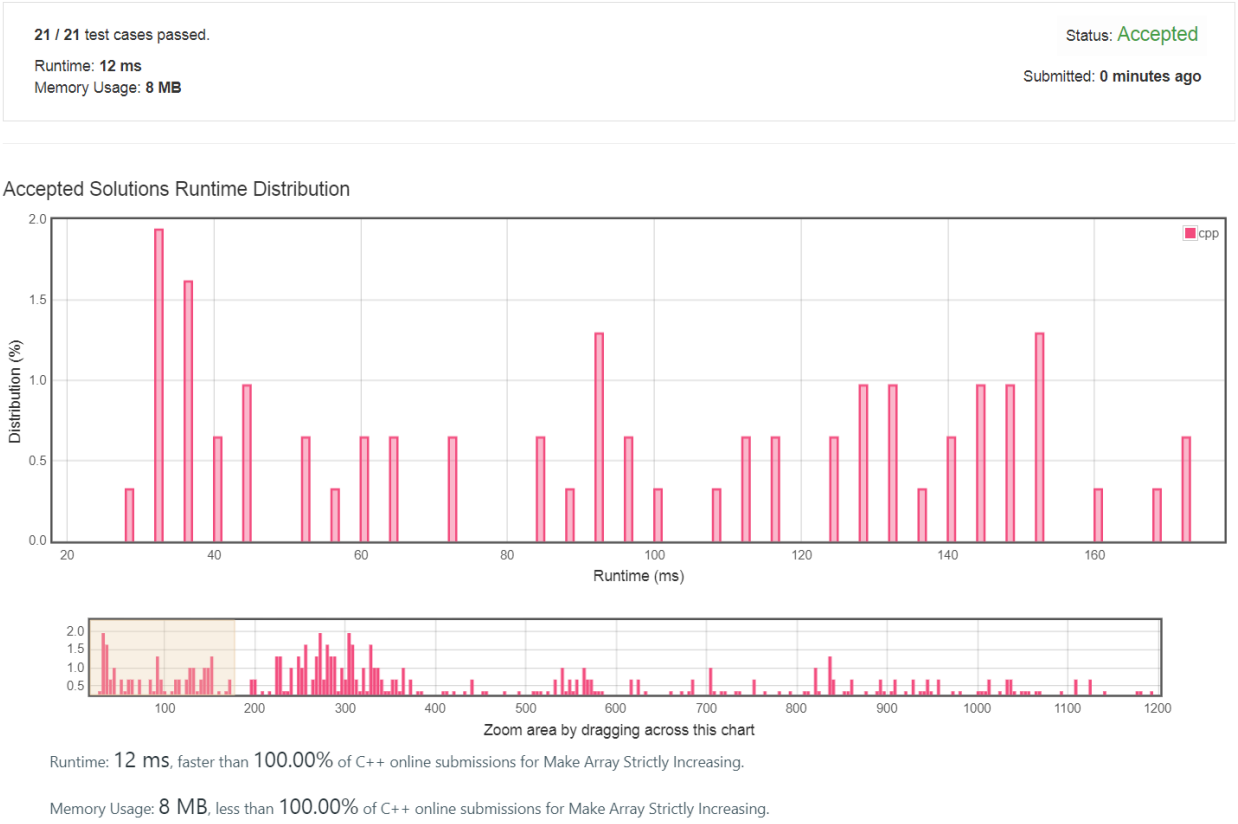
We can reduce the problem to 2D LIS. Let $r[i]$ denote the rank of $a[i]$ in array b (a minor issue is when $\exists\ j$ s.t. $a[i] = b[j]$), then the problem is equivalent to find the longest increasing subsequence in a, where $a[i]$ can be followed by $a[j]$ if $a[i] < a[j]$ and we can change $a[i+1, \ldots, j-1]$ to elements in b, which is equivalent to $r[j] - r[i] \geq j - i - 1$, i.e. $r[j] - j \geq r[i] - i - 1$ (the -1 shouldn't affect much). cite?
Similar to the 1D LIS solution, we can dynamically maintain a set of disjoint staircase structures in 2D, where the i-th staircase represents the points with LIS length i ending at it. $O(\frac{\log n}{\log \log n})$ using dynamic planar orthogonal point location [1].



References

[1] Timothy M Chan and Konstantinos Tsakalidis. Dynamic planar orthogonal point location in sublogarithmic time. In *34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.