

1. optimize DP by binary search.  $O(n \log n)$ .
2. there's an  $O(\text{sort}(n) + n \log \log n)$  time algorithm [3] using vEB trees, and also  $O(n \log \log n)$  [1] for computing LIS of a permutation with  $n$  numbers.

lower bound:  $\Omega(n \log n)$  in comparison-based model [2].

## References

- [1] Maxime Crochemore and Ely Porat. Fast computation of a longest increasing subsequence and application. *Information and computation*, 208(9):1054–1059, 2010.
- [2] Michael L Fredman. On computing the length of longest increasing subsequences. *Discrete Mathematics*, 11(1):29–35, 1975.
- [3] James W Hunt and Thomas G Szymanski. A fast algorithm for computing longest common subsequences. *Communications of the ACM*, 20(5):350–353, 1977.