LIS in n log n time

Wednesday, 28 August 2024 10:37 PM

Input: Sequence of n distinct integers
$x_1, x_2, \ldots, x_n$

Output: Largest subset $S \subseteq [n]$ such that if
$i, j \in S$ & $i < j$, then $x_i < x_j$
(i.e., indices for the longest increasing subsequence)

Algorithm Quick LIS:

array $M[1 \ldots n] = (\infty, \ldots, \infty)$
length $\ell = 0$

for $i = 1 \ldots n$
  let k be smallest index s.t. $M(k) > x_i$
  if $(k > \ell)$
    $\ell = \ell + 1$, $M(\ell) = x_i$
  else
    $M(k) = x_i$

Example: 6, 12, 4, 3, 7, 10, 8, 18, 11, 16, 9

ℓ = ~~1~~ ~~2~~ 3 4 5

M: ~~6~~ ← ~~12~~ ~~10~~ ~~18~~ 16
~~4~~ 7 ← 8 ← ~~11~~
3 9

Algorithm clearly runs in $O(n \log n)$ time.

Claim: For any index k, $M(k)$ decreases every time it changes.
(easy)

Claim: In any iteration i,
$M(1) \leq M(2) \leq \ldots \leq M(n)$
(easy)

Claim: After any iteration i, for $k \in \{1, \ldots, n\}$
(i) If $M(k) < \infty$, $M(k)$ stores the smallest value in $x_1, \ldots, x_i$ s.t. there is an increasing subsequence of length k that ends in $M(k)$
i.e., every subsequence of length k in $x_1, \ldots, x_i$ terminates in a value $\geq M(k)$
(ii) there is an increasing subsequence of length ℓ

Proof: By induction.
Base case: true after first iteration.
Suppose true after i iterations.
Say k is the smallest index s.t. $M(k) > x_{i+1}$
Then $M(1), \ldots, M(k-1) < x_{i+1} < M(k), M(k+1), \ldots$

- for $r \leq k-1$,
  claim remains true.
- for $r = k$,
  there is a size k IS that ends in x_{i+1}
  & no size k IS ends in a smaller value
- for $r > k$, note that $M(r-1) > x_{i+1}$ earlier
  if $M(r) = \infty$
    then there was no size r IS in $x_1, \ldots, x_i$
    if there now is a size r IS in $x_1, \ldots, x_{i+1}$
      x_{i+1} is last elt.
    but then there was a size $r-1$ IS in $x_1, \ldots, x_i$
      ending in $v < x_{i+1}$
    & $M(r-1) < x_{i+1}$, contradiction
  if $M(r) < \infty$
    then $M(r)$ is the smallest last value for any size r IS in $x_1, \ldots, x_i$
    ...
(complete yourself)