

② LIS问题,

Date. 2019 Page. 1000

$a[5] = [1, 5, 3, 4, 8]$, 求最长严格↑子序列;

方法 - ($O(n^2)$).

① $b[5] = \text{单增排序}(a[5]) = [1, 3, 4, 5, 8]$

② $D[i][j]$ 是 a 从 $0 \sim i-1$, b 从 $0 \sim j-1$ 的最长公共子序列长度

③ $D[i][j] = \begin{cases} 0, & i=0, \text{ or } j=0 \\ D[i-1][j-1] + 1, & a[i-1] = b[j-1] \\ \max(D[i-1][j], D[i][j-1]), & a[i-1] \neq b[j-1] \end{cases}$

④

	0	1	3	4	5	8
0	0	0	0	0	0	0
1	0	1	1	1	1	1
5	0	1	1	1	2	2
3	0	1	2	2	2	2
4	0	1	2	3	3	3
8	0	1	2	3	3	4

路径为

1 3 4 8
最长公共子序列长度为 4

⑤ 伪代码.

$D[i][0] = 0, D[0][j] = 0.$

for ($i=1, i \leq 5, i++$)

for ($j=1, j \leq 5, j++$)

if $a[i-1] = b[j-1]$

$D[i][j] = D[i-1][j-1] + 1$

else

$D[i][j] = \max \begin{cases} D[i-1][j] \\ D[i][j-1] \end{cases}$

return $D[5][5]$

② 方法 = $O(n^2)$,

$A[5] = [1, 5, -3, 4, 8]$

① $PC[i]$ 是 A 从 $0 \sim i$ 的最长 ↑ 子序列长度

$Q[i]$ 是 储存 $PC[i]$ 的上一个元素

② $PC[0] = 1$, $i \geq 1$ 时, $PC[i] = \max\{ \overset{x < i \text{ 且}}{A[x] < A[i]} ? PC[x] : 0 \} + 1$

$x < i$ 且 $A[x] < A[i]$ 且 $PC[x]$ is max, $Q[i] = A[x]$. 不存在时 $Q[i] = A[i]$

③ $A[5] = [1, 5, -3, 4, 8]$

$PC[5] = [1, 2, 1, 2, 3]$

$Q[5] = [1, 1, -3, 1/-3, 5/4]$

$\therefore \max \text{ len} = 3,$

序列为, $8 \rightarrow 5 \rightarrow 1$
 $\quad \quad \quad \rightarrow 4 \rightarrow 1$
 $\quad \quad \quad \quad \rightarrow -3$

④ 伪代码

方法三 ($O(n \log n)$)

$A[8] = [1, 4, 7, 2, 5, 9, 10, 3]$

① 新建数组 $B[8] = [0, 0, \dots, 0]$, $ans = 0$

② 把 $A[0]$ 放入, $B[0] = 1$, $ans = 1$

把 $A[i]$ 放入 $\begin{cases} A[i] \text{ 比前面数都大, } B[i] = A[i], \text{ ans}++; \\ \text{从左向右, } A[i] \text{ 替换第一个比它大的数} \end{cases}$

③ $B[8] = [1, 4, 7, 9, 10,] = [1, 2, 3, 9, 10]$
 2 3
 3
 $ans = 5;$

④ $\max \text{ len} = \text{len}$ ~~ans~~ $ans = 5;$

⑤ 但结果 $[1, 2, 3, 9, 10]$ 不是严格 ↑ 子序列;
 实际上, 这种算法得出的 ans 是严格 ↑ 子序列的 ^{max} 长度
 $B[]$ 可能是最长严格 ↑ 子序列.
 也可能不是, 但 $B[]$ 会一直更新;

找第一个 $\geq x$ 的位置

⑥ 伪代码: $\text{int binary-search}(A \text{ 数组, 长度 } R, x), \text{ int main}()$

```

int L = 1, mid;
while (L ≤ R)
    mid = (L + R) >> 1;
    if (A[mid] ≤ x)
        L = mid + 1;
    else
        R = mid - 1;
return L;

B[0] = A[0], ans = 1;
for (int i = 1; i < 8; i++)
    { if A[i] > B[ans]
      B[ans] = A[i];
      ans++;
    }
else,
    B[binary-search(B, ans, A[i])] = A[i];
print f(ans)
  
```