

LIS

- 思路 1. 记 $a[i]$ 为第 i 个元素结尾的最长递增子序列
 $O(n^2)$ 并定义 $\text{parent}[i]$, $\text{parent}[i]$ 记录在 $A[i]$ 之前的元素的索引.
2. 若 $A[j] < A[i]$ 且 $a[j]+1 > a[i]$, 则: $a[i]=a[j]+1$
 $\text{parent}[i]=j$
3. 遍历 $a[i]$, 找到最大值 并根据 $\text{parent}[i]$ 回溯路径.

伪代码: 输入: 数组 A , 长度 n .

1. 初始化 $a[i]=1$, $\text{parent}[i]=-1$ (对于所有 $i \in [0, n-1]$)
2. for $i=1$ to $n-1$ do:
 for $j=0$ to $i-1$ do:
 if $A[j] < A[i]$ and $a[j]+1 > a[i]$:
 $a[i]=a[j]+1$, $\text{parent}[i]=j$

3. 找到 a 中的最大值 \max 及索引 \max_index .

4. 初始化 $\text{result} = []$

从 \max_index 开始, 重复:

 将 $A[\max_index]$ 加入 result

$\max_index = \text{parent}[\max_index]$

 直到 $\max_index == -1$.

5. 返回 \max 及反转后的 result

eg: 输入 $A = [10, 22, 9, 33, 21, 50, 41, 60]$, $n=8$

① $a = [1, 1, 1, 1, 1, 1, 1, 1]$ parent = [-1, -1, -1, -1, -1, -1, -1, -1]

i=1: $j=0$. $A[0] < A[1]$ 且 $a[0]+1 > a[1]$ 则 $a[1]=a[0]+1=2$, parent[1]=0

j=2: $j=0, 1$ 均不满足 $A[j] < A[1]$, 不变.

i=3: $j=0$, $A[0] < A[3]$ 且 $a[0]+1 > a[3]$. 则 $a[3]=2$, parent[3]=0

j=1, $A[0] < A[3]$ 且 $a[1]+1 > a[3]$. 则 $a[3]=3$, parent[3]=-1

j=2, $A[2] < A[3]$ 但 $a[2]+1 < a[3]$. 不变

i=4: $j=0$. $A[0] < A[4]$, 且 $a[0]+1 > a[4]$. $\therefore a[4]=2$, parent[4]=0

j=1, $A[1] > A[4]$. j=2, $A[2] < A[4]$ 但 $a[2]+1 = a[4]$, 不变

j=3, $A[3] > A[4]$.

i=5: $j=0$. $A[0] < A[5]$. $a[0]+1 > a[5]$. $\therefore a[5]=2$. parent[5]=0

j=1. $A[1] < A[5]$, $a[1]+1 > a[5]$. $\therefore a[5]=3$. parent[5]=-1

j=2. $A[2] < A[5]$ 但 $a[2]+1 < a[5]$. j=3. 可得 $a[5]=4$. parent[5]=3

③ 以^上类推, 可得 $a = [1, 2, 1, 3, 2, 4, 3, 5]$, max=5, max_index=7
parent = [-1, 0, -1, 1, 0, 3, 3, 5]

④ 回溯: 从索引 7 开始, 沿 parent 回溯 $7 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow -1$

$60 \rightarrow 50 \rightarrow 33 \rightarrow 22 \rightarrow 10$

⑤ 反转得到 LIS: $[10, 22, 33, 50, 60]$

$O(n \log n)$

- 思路：1. 用 $a[i]$ 来记录当前长度为 k 的递增子序列的最小末尾元素。
parent[i] 记录 LIS 的路径，parent[i] 表示当前元素之前索引
position[i]：记录当前 LIS 长度中末尾元素的索引。
2：对当前 $A[i]$ ，二分查找 找到 $a[i]$ 中第一个大于 $A[i]$ 的位置 pos。
更新 $a[pos]$
3：从记录的最长 LIS 的最后一个元素开始，利用 parent[i] 回溯。

伪代码：输入数组 a ，长度 n 。

1. 初始化空数组 a 、parent[i] = -1
2. 初始化 position[i] 记录每个长度的末尾元素的索引
3. for $i=0$ to $n-1$ do：
 使用二分查找 找到 a 中第一个大于等于 $A[i]$ 的位置 pos
 if $pos == a$ 的长度：
 将 $A[i]$ 加到 a 的末尾

or :

更新 $a[pos] = A[i]$

记录 $parent[i] = position[pos-1]$ (if $pos > 0$)

更新 $position[pos] = i$

4. 从 position 最后一个元素开始 回溯路径。

5. 返回 a 的长度及反转的路径

eg: 输入 $A = [10, 22, 9, 33, 21, 50, 41, 60]$, $n = 8$

① ~~a = []~~ $a = []$, $\text{parent} = [-1, 1, 1, -1, -1, -1, -1]$, $\text{position} = []$

② $A[0] = 10$ { 二分查找: a 为空, 直接将 10 插入

更新: $a = [10]$

记录: $\text{parent}[0] = -1$, $\text{position}[0] = 0$

$A[1] = 22$ { 二分查找: $22 > 10$, 插入到 a 末尾, $a = [10, 22]$

$\text{parent}[1] = 0$, $\text{position}[1] = 1$

$A[2] = 9$ { 二分查找: $9 < 10$, 替换 $a[\text{pos}] = a[0]$, $a = [9, 22]$

$\text{parent}[2] = -1$, $\text{position}[0] = 2$

$A[3] = 33$ { 二分查找: $33 > 22$, 插入到 a 末尾, $a = [9, 22, 33]$

$\text{parent}[3] = 1$, $\text{position}[2] = 3$

$A[4] = 21$ { $21 < 22$, 替换 $a[1]$, $a = [9, 21, 33]$

$\text{parent}[4] = 2$, $\text{position}[1] = 4$

$A[5] = 50$ { $50 > 33$, 插入到 a 末尾, $a = [9, 21, 33, 50]$

$\text{parent}[5] = 3$, $\text{position}[3] = 5$

③ $A[6], A[7]$ 同理, 最后 $a = [9, 21, 33, 41, 60]$

$\text{parent} = [-1, 0, -1, 1, 2, 3, 3, 6]$, $\text{position} = [2, 4, 3, 6, 7]$

④ 回溯: $\text{position}[4] = 7$, $A[7] = 60$ |. $\text{parent}[7] = 6 \rightarrow A[6] = 41$

$\text{parent}[6] = 3 \rightarrow A[3] = 33$ |. $\text{parent}[3] = 1 \rightarrow A[1] = 22$

$\text{parent}[1] = 0 \rightarrow A[0] = 10$

得到 LIS: $[10, 22, 33, 41, 60]$