LIS(单调递增数列) 实现

梁育玮 3230102923

2024 年 12 月 16 日

I. 大致思路

维护一个大小不超过 n 的一维数组 s 和他的大小 size_ 以及一个数组 b，$s[i]$ 表示长度为 i 的递增数列里面，末值的最小可能，保证 s 是升序的。$b[i]$ 表示第 i 个元素结尾的递增序列最大长度。

每读入一个数 $a[i]$，记 j 是满足 $s[j] < a[i]$ 的最大下标，将 $s[j+1]$ 设为 t，$b[i]$ 设为 j+1，同时视情况更新 size 的值，最终数组 s 的长度即为 LIS 序列的最大长度

最后通过数组 b 倒着找满足要求的序列

伪代码

下面是伪代码

```
a is a given array
n = a.size
b = an array of size n
s = an array of size n+1
size_ = 1
s[1] = a[0]
b[0] = 1
for i = 1 to n-1
    //bijection search
    pos = 0
    l, r = 1, size
    while l <= r
        mid = (l + r)/2
        if a[i] > s[mid]
            pos = mid
```

```
            l = mid + 1
        else r = mid - 1
    s[pos + 1] = a[i]
    b[i] = pos + 1
m, j = size_, n-1
LST = an array of size size_
//print the LIS
while m > 0
    while b[j] != m
        j = j -1
    LST[m-1] = a[j]
    m = m - 1
```

同文件夹下的'1.py' 中有根据伪代码自动生成的 python 代码，python 代码中有对接下来展示的数据的详细处理

II. 例子

给定一列数据：a = [2, 1, 4, 3, 2, 5, 6, 3, 2]

- 处理 2：将 s[1] 设为 2，size 设为 1，b[0] = 1
- 处理 1：没有比 1 小的，将 s[1] 设为 1，b[1] = 1
- 处理 4：1 比 4 小，将 s[2] 设为 4，同时更新 size_ 为 2，b[2] = 2
- 处理 3：1 比 3 小，将 s[2] 设为 3，b[3] = 2
- 处理 2：1 比 2 小，将 s[2] 设为 2，b[4] = 2
- 处理 5：2 比 5 小，将 s[3] 设为 5，同时更新 size_ 为 3，b[5] = 3
- 处理 6：5 比 6 小，将 s[4] 设为 5，同时更新 size_ 为 4，b[6] = 4
- 处理 3：2 比 3 小，将 s[3] 设为 3，b[7] = 3
- 处理 2：1 比 2 小，将 s[2] 设为 2（实际上没有变化），b[8] = 2
- size = 4 为数组长度，b=[1, 1, 2, 2, 2, 3, 4, 3, 2]
- 最后找到要求的 LIS

时间复杂度分析

二分查找的时间复杂度为 $\Theta(\log \text{size\_})$，故构建数组 s 和 b 的时间复杂度是 $\mathcal{O}(n \log n)$，构建 LST 的时间复杂度为 $\mathcal{O}(n)$，故整体时间复杂度为 $\mathcal{O}(n \log n)$