

{% note info %} **摘要** Title: 802. 区间和 Tag: 离散化 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:NEFU AB-IN

[Link](#)

@TOC

802. 区间和

• 题意

假定有一个无限长的数轴，数轴上每个坐标上的数都是 0。现在，我们首先进行 n 次操作，每次操作将某一位置 x 上的数加 c 。接下来，进行 m 次询问，每个询问包含两个整数 l 和 r ，你需要求出在区间 $[l,r]$ 之间的所有数的和。

• 思路

一开始的思路是

- 将需要操作的数加入离散化数组进行离散，通过二分操作查询到坐标，操作到答案数组
- 答案数组进行前缀和操作
- 查询的 l,r 也进行二分查找

注意：

- 离散化数组最好加上 $**-INF$ 和 $INF**$ ，方便处理边界问题
- 是要把所有需要离散的值放进离散化数组
- 不要忘了存原数组

• 代码

```
...
Author: NEFU AB-IN
Date: 2022-02-23 16:11:57
FilePath: \ACM\Acwing\802.py
LastEditTime: 2022-02-23 16:41:23
...

N = int(1e5 + 100)
b = [0] * N
a = []
xs = []
INF = int(2e9)

def findl(x): #返回第一个大于等于x的下标，同lower_bound
    l, r = 0, len(xs) - 1
    while l < r:
```

```
        mid = l + r >> 1
        if xs[mid] >= x:
            r = mid
        else:
            l = mid + 1
    return r

def findr(x): #返回最后一个小于等于x的下标, 不同upper_bound
    l, r = 0, len(xs) - 1
    while l < r:
        mid = l + r + 1 >> 1
        if xs[mid] <= x:
            l = mid
        else:
            r = mid - 1
    return r

n, m = map(int, input().split())
for i in range(n):
    x, c = map(int, input().split())
    a.append([x, c])
    xs.append(x)

xs = [-INF, *xs, INF]
xs = list(set(xs))
xs.sort()

for i in range(len(a)):
    t = a[i][0]
    L = findl(t)
    b[L] += a[i][1]

for i in range(1, len(b)):
    b[i] += b[i - 1]

for i in range(m):
    l, r = map(int, input().split())
    L = findl(l)
    R = findr(r)
    print(b[R] - b[L - 1])
```