重庆南开中学

离散化思想

南开中学信息学竞赛教练组

先看个例题

例题【好多狼骑 NKOJ1178】：

食尸鬼探查清楚了qcy的兵力的情况，它发现qcy的基地里有好多的狼骑。但是由于食尸鬼的智商有限，它记不下太大的数字，只能分成n次看，记下位置ai到bi位置段有无狼骑。

输入格式：

第一行一个数n表示食尸鬼看了n次
接下来第二到n+1行，两个每行数ai,bi，表示第ai到bi的位置站满了狼骑(一个单位坐标站一只)。
数据范围：1<=N<=20000 ai<=bi<=9999999999

输出格式：

狼骑的总数x。

样例输入：
3
1 3
2 4
6 7

样例输出：
6

坐标	1	2	3	4	5	6	7
狼骑	1	1	1	1	0	1	1

共6只

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始 {false, false, false, false, false, false, false, false, false, ...}

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始 {false, false, false, false, false, false, false, false, false, ...}
  - 标记[1,3] {false, true , true , true , false, false, false, false, false, ...}

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始 {false, false, false, false, false, false, false, false, false, ...}
  - 标记[1,3] {false, true , true , true , false, false, false, false, false, ...}
  - 标记[2,4] {false, true , true , true , true , false, false, false, false, ...}

先来看个例题

重庆南开中学

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始 {false, false, false, false, false, false, false, false, false, ...}
  - 标记[1,3] {false, true , true , true , false, false, false, false, false, ...}
  - 标记[2,4] {false, true , true , true , true , false, false, false, false, ...}
  - 标记[6,7] {false, true , true , true , true , false, true , true , false, ...}

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始　{false, false, false, false, false, false, false, false, false, ...}
  - 标记[1,3]　{false, true , true , true , false, false, false, false, false, ...}
  - 标记[2,4]　{false, true , true , true , true , false, false, false, false, ...}
  - 标记[6,7]　{false, true , true , true , true , false, true , true , false, ...}
  - 统计：共有6只狼骑。

先来看个例题

重庆南开中学

例题【好多狼骑 NKOJ1178】：

解析：

- 算法1：
- 开一个bool类型数组，将有狼骑的坐标标记为true
  - 一开始　{false, false, false, false, false, false, false, false, false, ...}
  - 标记[1,3]　{false, true , true , true , false, false, false, false, false, ...}
  - 标记[2,4]　{false, true , true , true , true , false, false, false, false, ...}
  - 标记[6,7]　{false, true , true , true , true , false, true , true , false, ...}
  - 统计：共有6只狼骑。
- 时间/空间复杂度：
  - 对每个区间[ai,bi]，时间复杂度为区间长度bi-ai+1
  - 总时间复杂度$O(N \times Range)$，会超时。
  - 空闲复杂度$O(Range)$，存不下

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法2：差分思想
- 第一步，开一个整数数组x[1...Range]
    - 在a[i]的位置写个+1，表示“有一段狼骑开始啦”
    - 在b[i]+1的位置写个-1，表示“有一段狼骑结束啦”

先来看个例题

重庆南开中学

例题【好多狼骑 NKOJ1178】：

解析：

- 算法2：差分思想
- 第一步，开一个整数数组x[1...Range]
    - 在a[i]的位置写个+1，表示“有一段狼骑开始啦”
    - 在b[i]+1的位置写个-1，表示“有一段狼骑结束啦”
    - 例如[1,3],[2,4],[6,7]如下所示

数组下标i	1	2	3	4	5	6	7	8
数组数值x[i]	+1	+1	0	-1	-1	+1	0	-1

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法2：差分思想
- 第一步，开一个整数数组x[1...Range]
  - 在a[i]的位置写个+1，表示“有一段狼骑开始啦”
  - 在b[i]+1的位置写个-1，表示“有一段狼骑结束啦”
  - 例如[1,3],[2,4],[6,7]如下所示
- 第二步，求x[]数组的前缀和，统计有多少个位置＞0
  - 如右图所示，共有6个位置＞0，所以答案是6

数组下标i	1	2	3	4	5	6	7	8
数组数值x[i]	+1	+1	0	-1	-1	+1	0	-1
求前缀和	1	2	2	1	0	1	1	0

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法2：差分思想
- 第一步，开一个整数数组x[1...Range]
    - 在a[i]的位置写个+1，表示“有一段狼骑开始啦”
    - 在b[i]+1的位置写个-1，表示“有一段狼骑结束啦”
    - 例如[1,3],[2,4],[6,7]如下所示
- 第二步，求x[]数组的前缀和，统计有多少个位置＞0
    - 如右图所示，共有6个位置＞0，所以答案是6
- 时间/空间复杂度？
    - 时间复杂度：第一步$O(N)$，第二步$O(Range)$，仍然会超时，但比算法1好很多了
    - 空间复杂度$O(Range)$，仍然存不下。

数组下标i	1	2	3	4	5	6	7	8
数组数值x[i]	+1	+1	0	-1	-1	+1	0	-1
求前缀和	1	2	2	1	0	1	1	0

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + <u>离散化思想</u>
- 观察算法2，数组x[1..Range]是主要瓶颈

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + <u>离散化思想</u>
- 观察算法2，数组x[1..Range]是主要瓶颈
- 如果不开这个数组，而是将操作记录下来
  - x[a[i]]++ 改成 (a[i], +1)
  - x[b[i]+1]-- 改成 (b[i]+1, -1)

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
```

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 观察算法2，数组x[1..Range]是主要瓶颈
- 如果不开这个数组，而是将操作记录下来
  - x[a[i]]++ 改成 (a[i], +1)
  - x[b[i]+1]-- 改成 (b[i]+1, -1)
  - 将这2N个“(位置, ±1)”形式的数据按位置排序

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
```

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 ＋ <u>离散化思想</u>
- 统计答案依靠±1的前缀和：

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:      1       2       1       0       1       0
```

先来看个例题

重庆南开中学

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1

[1,3]:	(1,+1),(4,-1)					
[2,4]:	(2,+1),(5,-1)					
[6,7]:	(6,+1),(8,-1)					
sort :	(1,+1)	(2,+1)	(4,-1)	(5,-1)	(6,+1)	(8,-1)
前缀和:	1	2	1	0	1	0

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + <u>离散化思想</u>
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
  - 位置区间[2,4)，前缀和为2。 ans += 4 - 2

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:      1       2       1       0       1       0
```

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
  - 位置区间[2,4)，前缀和为2。 ans += 4 - 2
  - 位置区间[4,5)，前缀和为1。 ans += 5 - 4

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:     1       2       1       0       1       0
```

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
  - 位置区间[2,4)，前缀和为2。 ans += 4 - 2
  - 位置区间[4,5)，前缀和为1。 ans += 5 - 4
  - 位置区间[5,6)，前缀和为0。

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:      1       2       1       0       1       0
```

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
  - 位置区间[2,4)，前缀和为2。 ans += 4 - 2
  - 位置区间[4,5)，前缀和为1。 ans += 5 - 4
  - 位置区间[5,6)，前缀和为0。
  - 位置区间[6,8)，前缀和为1。 ans += 8 - 6

[1,3]:	(1,+1),(4,-1)					
[2,4]:	(2,+1),(5,-1)					
[6,7]:	(6,+1),(8,-1)					
sort :	(1,+1)	(2,+1)	(4,-1)	(5,-1)	(6,+1)	(8,-1)
前缀和:	1	2	1	0	1	0

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + 离散化思想
- 统计答案依靠±1的前缀和：
  - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
  - 位置区间[2,4)，前缀和为2。 ans += 4 - 2
  - 位置区间[4,5)，前缀和为1。 ans += 5 - 4
  - 位置区间[5,6)，前缀和为0。
  - 位置区间[6,8)，前缀和为1。 ans += 8 - 6
  - 最终ans值为6。

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:      1       2       1       0       1       0
```

重庆南开中学

先来看个例题

例题【好多狼骑 NKOJ1178】：

解析：

- 算法3：差分思想 + <u>离散化思想</u>
- 统计答案依靠±1的前缀和：
    - 位置区间[1,2)，前缀和为1。 ans += 2 - 1
    - 位置区间[2,4)，前缀和为2。 ans += 4 - 2
    - 位置区间[4,5)，前缀和为1。 ans += 5 - 4
    - 位置区间[5,6)，前缀和为0。
    - 位置区间[6,8)，前缀和为1。 ans += 8 - 6
    - 最终ans值为6。
- 时间复杂度：$O(N\log N)$
- 空间复杂度：$O(N)$

```
[1,3]:  (1,+1),(4,-1)
[2,4]:  (2,+1),(5,-1)
[6,7]:  (6,+1),(8,-1)
sort :  (1,+1)  (2,+1)  (4,-1)  (5,-1)  (6,+1)  (8,-1)
前缀和:      1       2       1       0       1       0
```

再来看个例题

重庆南开中学

例题【图形面积 NKOJ1353】：

桌面上放了N个矩形，这N个矩形可能有互相覆盖的部分，求它们组成的图形的面积。(矩形的边都与坐标轴平行)。

输入格式：

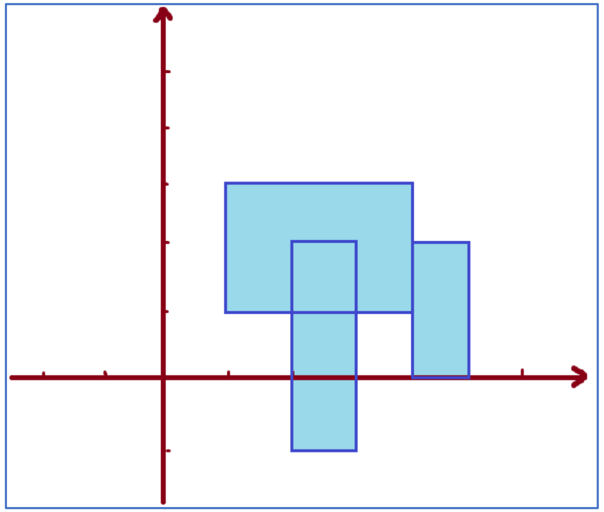
输入第一行为一个数N（1⩽N⩽100），表示矩形的数量。
下面N行，每行四个整数，分别表示每个矩形的左下角和右上角的坐标。
坐标范围为-10^8到10^8之间的整数。

输出格式：

输出只有一行，一个整数，表示图形的面积。

样例输入：
3
1 1 4 3
2 -1 3 2
4 0 5 2

样例输出：
10



再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法1：
- 坐标都是整数，所以开一个二维数组来标记每个1×1小方格是否被覆盖！
  - 需要先将所有矩形，平移到第一象限
- 坐标范围非常大，时间和空间复杂度都太高。

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法2：离散化
- 坐标范围很大，但出现至多$2N$个不同的横坐标，$2N$个不同的纵坐标

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

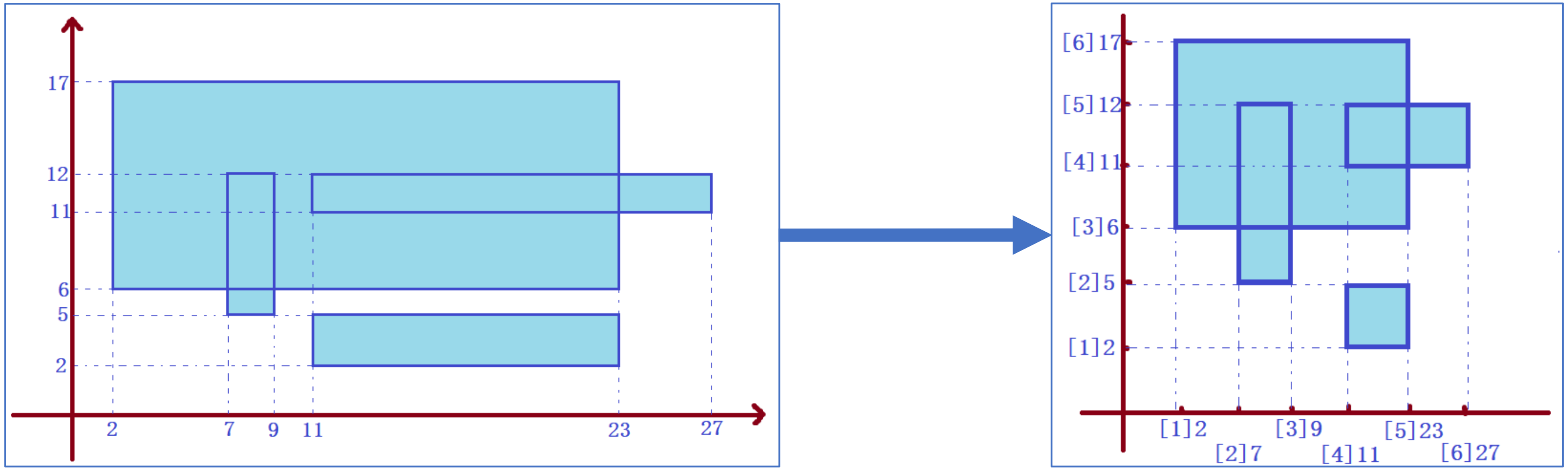
- 算法2：离散化
- 坐标范围很大，但出现至多$2N$个不同的横坐标，$2N$个不同的纵坐标
  - 将这些坐标排序去重后，用数组下标代替原来的坐标
  - 新的坐标（数组下标）范围在$[1,2N]$以内

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法2：离散化
- 坐标范围很大，但出现至多$2N$个不同的横坐标，$2N$个不同的纵坐标
  - 将这些坐标排序去重后，用数组下标代替原来的坐标
  - 新的坐标（数组下标）范围在$[1,2N]$以内



再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法2：离散化
- 离散化之后，套用算法1，二维数组统计每个“1×1正方形”是否被覆盖

重庆南开中学

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

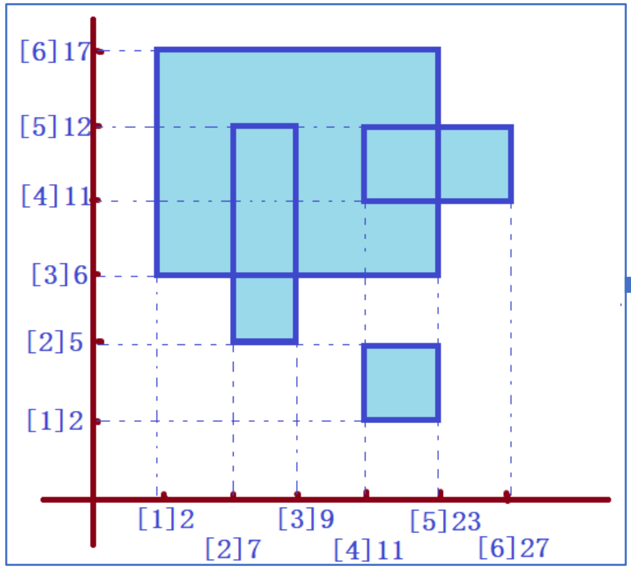
- 算法2：离散化
- 离散化之后，套用算法1，二维数组统计每个“1×1正方形”是否被覆盖
  - 每个正方形，暴力标记数组，时间复杂度$O(2N \times 2N)$
  - 注意每个“1×1正方形”，原本实际是个矩形

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法2：离散化
- 离散化之后，套用算法1，二维数组统计每个“1×1正方形”是否被覆盖
  - 每个正方形，暴力标记数组，时间复杂度$O(2N\times 2N)$
  - 注意每个“1×1正方形”，原本实际是个矩形



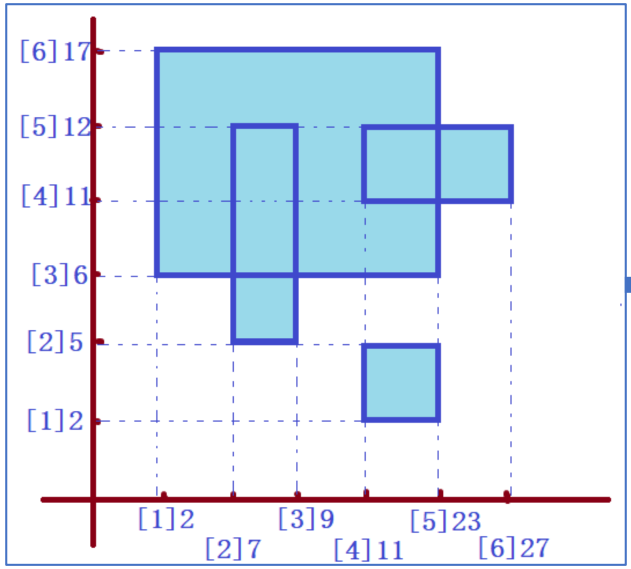

```
6 |
5 |  1  1  1  1
4 |  1  1  1  1  1
3 |  1  1  1  1
2 |     1
1 |           1
  +------------------
     1  2  3  4  5  6
```

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法2：离散化
- 离散化之后，套用算法1，二维数组统计每个“1×1正方形”是否被覆盖
  - 每个正方形，暴力标记数组，时间复杂度$O(2N \times 2N)$
  - 注意每个“1×1正方形”，原本实际是个矩形
- 总时间复杂度$O(N^3)$。




```
6 |
5 |  1  1  1  1
4 |  1  1  1  1  1
3 |  1  1  1  1
2 |     1
1 |           1
  +------------------
     1  2  3  4  5  6
```

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

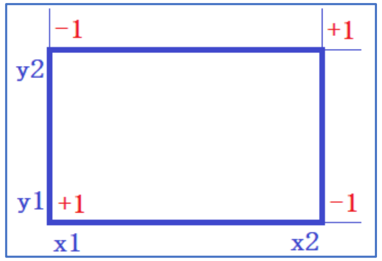
- 算法3：离散化 + 差分思想
- 二维差分：
  - 矩形[x1, x2]×[y1, y2]，其中x1, x2, y1, y2都是离散化后的坐标
    - a[x1][y2+1]--　　　　a[x2+1][y2+1]++
    - a[x1][y1]++　　　　a[x2+1][y1]--

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法3：离散化 + 差分思想
- 二维差分：
  - 矩形[x1,x2]×[y1,y2]，其中x1,x2,y1,y2都是离散化后的坐标
    - a[x1][y2+1]--　　　　a[x2+1][y2+1]++
    - a[x1][y1]++　　　　a[x2+1][y1]--

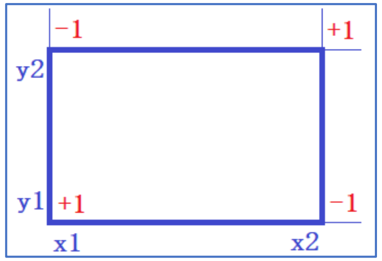


再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法3：离散化 ＋ 差分思想
- 二维差分：
  - 矩形[x1, x2]×[y1, y2]，其中x1, x2, y1, y2都是离散化后的坐标
    - a[x1][y2+1]--　　　　a[x2+1][y2+1]++
    - a[x1][y1]++　　　　a[x2+1][y1]--
  - 每个正方形标记四个点，时间复杂度$O(1)$，共$O(N)$

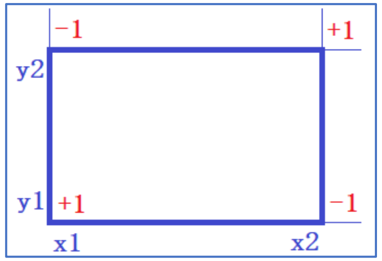


再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法3：离散化 + 差分思想
- 二维差分：
    - 矩形[x1,x2]×[y1,y2]，其中x1,x2,y1,y2都是<u>离散化后</u>的坐标
        - a[x1][y2+1]-- a[x2+1][y2+1]++
        - a[x1][y1]++ a[x2+1][y1]--
    - 每个正方形标记四个点，时间复杂度$O(1)$，共$O(N)$
- 二维前缀和：s[x][y] = s[x-1][y] + s[x][y-1] - s[x-1][y-1] + a[x][y]

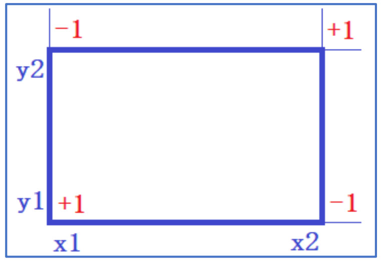


再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法3：离散化 + 差分思想
- 二维差分：
  - 矩形[x1, x2]×[y1, y2]，其中x1, x2, y1, y2都是离散化后的坐标
    - a[x1][y2+1]--　　　　　a[x2+1][y2+1]++
    - a[x1][y1]++　　　　　　a[x2+1][y1]--
  - 每个正方形标记四个点，时间复杂度$O(1)$，共$O(N)$
- 二维前缀和：s[x][y] = s[x-1][y] + s[x][y-1] - s[x-1][y-1] + a[x][y]




```
6 | -1   -1   +1        +1
5 |                -1   +1
4 |                +1   -1
3 | +1                  -1
2 |      +1   -1        -1   +1
1 |                     +1   -1
  +------------------------------
     1    2    3    4    5    6
```

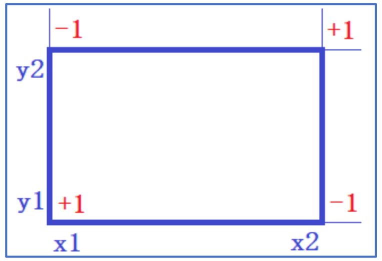
```
6 |
5 |  1    1    1    1
4 |  1    2    1    2    1
3 |  1    2    1    1
2 |       1
1 |                     1
  +------------------------------
     1    2    3    4    5    6
```

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法3：离散化 + 差分思想
- 二维差分：
  - 矩形[x1, x2]×[y1, y2]，其中x1, x2, y1, y2都是离散化后的坐标
    - a[x1][y2+1]--　　　　a[x2+1][y2+1]++
    - a[x1][y1]++　　　　a[x2+1][y1]--
  - 每个正方形标记四个点，时间复杂度$O(1)$，共$O(N)$
- 二维前缀和：s[x][y] = s[x-1][y] + s[x][y-1] - s[x-1][y-1] + a[x][y]
  - 统计答案和算法2类似
- 总时间复杂度$O(N^2)$。




```
6 | -1  -1  +1      +1
5 |             -1  +1
4 |             +1  -1
3 | +1              -1
2 |     +1  -1      -1  +1
1 |                 +1  -1
  +------------------------
     1   2   3   4   5   6
```

```
6 |
5 |  1   1   1   1
4 |  1   2   1   2   1
3 |  1   2   1   1
2 |      1
1 |                  1
  +------------------------
     1   2   3   4   5   6
```

再来看个例题

例题【图形面积 NKOJ1353】：

解析：

- 算法4：离散化 + 线段树
- 总时间复杂度$O(N\log N)$，学过线段树的同学们自行领悟。

总结一下

重庆南开中学

离散化思想：

- 原本数据范围很大，但关键数据不多
  - 原始数据范围可能是10^9、$\pm 10^{18}$，甚至浮点数、字符串
  - 关键数据一般只有$O(n)$个
- 通过排序等方式，只针对关键数据进行计算
  - 可以用作数组下标
  - 可以对相邻两个关键数据的区间进行整体计算

重庆南开中学

课后练习

习题

基础习题

NKOJ1178	好多狼骑
NKOJ1353	图形面积
NKOJ1377	火烧赤壁
NKOJ1279	岛屿

顺便复习

NKOJ2105	水晶球
NKOJ2102	烦人的生物

重庆南开中学

系统首页 | CQNK
oi.nks.edu.cn
Administrator
系统首页
练习题库
考试列表
判题结果
问题讨论与解答
统计信息与排名
用户中心
退出系统
Go
中文
日本語
English
首页
公告管理
编程的乐趣:创造和探索 (2019-01-10)
2019，一起奋斗，同学们加油！ (2019-01-01)
南开中学
Chongqing Nankai Secondary School
信息学竞赛
在线判题系统
Powered by
TOUCH STONE
本月解题排行
用户名 解题数
1 hdhd 47
2 Bot_Lee 40
3 Newuser 39
4 lukas0007 21
5 xiaojjj 20
编译器版本
GCC 7.3.0 (I/O: lld)
G++ 7.3.0 (I/O: lld)
JAVAC 1.8.0_31 (主类名: Main)
Free Pascal Compiler 2.6.4
Microsoft C# 4.0
默认不开启任何编译优化选项
常用IDE及工具下载
系统维护记录

Noi

oi.nks.edu.cn