题目翻译

1. 建筑4

题目描述：

奥林匹克运动会即将在乔伊王国举行。为了欢迎来自世界各地的参与者，从机场到住宿地的路上的建筑物都将被装修。有2N个从机场到住处的建筑物，从机场开始编号1至2N。

K总统负责装修工程。他要求公众制定装修计划。经过仔细研究，他最终选择了两个方案：方案A和方案B。在方案A中，i号楼（1≤i≤2N）的豪华程度为Ai，在方案B中，i号楼（1≤i≤2N）的豪华豪华为Bi。

两个计划都很好，很难选择其中一个。他决定用如下方式进行装修：对于每栋建筑，将选择一个方案A或方案B。为了公平地装修建筑物，将选择N栋建筑使用方案A，其余N栋建筑使用方案B。此外，因为如果在从机场到住处的途中豪华程度不断增加，参与者会很兴奋，所以应满足以下条件：对于1≤i≤2n-1的每个i，Ci≤C{i+1}，其中Ci是建筑i的豪华程度

写一个程序，给出建筑数量和每个计划的豪华程度，决定是否可以选择满足上述条件的装修方案。如果可能的话，输出一种装饰建筑物的方式。

题目输入：

N  
A1 A2 .... A{2N}

B1 B2 .... B{2N}

所以输入均为整数

题目输出：

如果不可能有合法装饰方案，输出-1。

否则，输出任意一个合法的2N长度的字符串S，其中第i位字符为A表示第i座建筑选择A方案，为B表示选择B方案。

数据限制：

1 ≤ N ≤ 500 000．

1 ≤ Ai ≤ 1 000 000 000 (1 ≤ i ≤ 2N).

1 ≤ Bi ≤ 1 000 000 000 (1 ≤ i ≤ 2N).

子任务：

1. (11 points) N ≤ 2 000.

2. (89 points) No additional constraints.

样例一：

输入：

3

2 5 4 9 15 11

6 7 6 8 12 14

输出：

AABABB

解释：

此方案中方案A和B均被选择3次，奢侈程度为2, 5, 6, 9, 12, 14满足限制

样例二：

输入：

2

1 4 10 20

3 5 8 13

输出：

BBAA

解释：

如果有多种方案，输出任意一种即可。

样例三：

输入：

2

3 4 5 6

10 9 8 7

输出：

-1

解释：

没有可行方案输出-1

样例四：

输入：

6

25 18 40 37 29 95 41 53 39 69 61 90

14 18 22 28 18 30 32 32 63 58 71 78

输出：

BABBABAABABA

2、汉堡牛排

题目描述：

你听说过奇幻发明有限公司吗？这家公司以“古怪的发明”而闻名。在这个问题中我们称之为JOI，Ltd。

一个新年派对正在JOI，Ltd.举行。工作人员正在一个巨大的铁丝网上烤N个汉堡牛排。我们将铁丝网视为一个1 000 000 000×1 000 000 000的网格。我们用（x，y）表示从左往右的第x列以及从下往上的第y行交汇的节点（1≤x≤1000 000 000，1≤y≤1000 000 000）

汉堡牛排的编号从1到N。汉堡牛排i（1≤i≤N）位于左下顶点为（Li，Di）右上顶点为（Ri，Ui）的矩形区域。汉堡牛排有可能重叠。

你是JOI，Ltd.的新员工，你的任务是在金属丝网上选择K个节点并在每个节点上面垂直地插上竹签。对于每一份汉堡牛排，你可以通过在它区域中插上至少一根竹签来确认它烹饪得如何了。你必须确认所有的汉堡牛排。一个节点可以多次插上竹签，你也可以在没有汉堡牛排的节点插上竹签。

形式上来说，您的任务是找到一个K对整数的元组（元素可以重复）（x1，y1）......（xK，yK）满足一下条件：

• 对于每一个i (1 ≤ i ≤ N)，都存在一个j (1 ≤ j ≤ K) 满足 Li ≤ xj ≤ Ri 同时Di ≤ yj ≤ Ui .

• 对于每一个j (1 ≤ j ≤ K)，都满足1 ≤ xj ≤ 1 000 000 000 和1 ≤ yj ≤ 1 000 000 000 .

写一个程序，给定汉堡牛排的位置和竹签的数量，计算一个插竹签的方法。输入保证存在至少一组解

题目输入：

N K

L1 D1 R1 U1

L2 D2 R2 U2

....

LN DN RN UN

所有值均为整数。

题目输出：

输出共K行，对于第j行(1 ≤ j ≤ K)，输出xj和yj，中间用空格隔开。如果有多组解，输出任意一组即可。

数据限制：

• 1 ≤ N ≤ 200 000.

• 1 ≤ K ≤ 4.

• 1 ≤ Li ≤ Ri ≤ 1 000 000 000 (1 ≤ i ≤ N).

• 1 ≤ Di ≤ Ui ≤ 1 000 000 000 (1 ≤ i ≤ N).

• There exists a tuple of K cells satisfying the conditions in the problem statement.

子任务：

1. (1 point) N ≤ 2000, K = 1.

2. (1 point) N ≤ 2000, K = 2.

3. (3 points) N ≤ 2000, K = 3.

4. (6 points) N ≤ 2000, K = 4.

5. (1 point) K = 1.

6. (3 points) K = 2.

7. (6 points) K = 3.

8. (79 points) K = 4.

样例一：

输入：

4 2

2 1 3 3

1 2 4 3

6 1 7 4

5 3 7 5

输出：

2 2

7 4

解释：

在(2,4)插入一根竹签，你可以确认1和2号牛排

在(7,4)插入一根竹签，你可以确认3和4号牛排

除了这种方案以外，(3,3)和(6,4)也成立。

样例二：

输入：

3 3

1 1 1 1

1 2 1 2

1 3 1 3

输出：

1 1

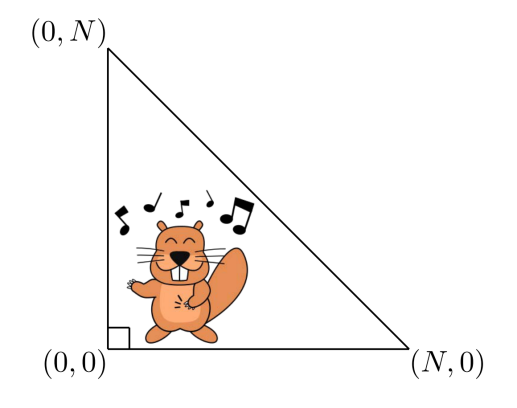
1 2

1 3

3、清扫

题目描述：

比塔罗的房间是等腰直角三角形，其直角边长为N。比塔罗房间的一个点可以表示为坐标（x，y），其中0≤x≤N，0≤y≤N，x+y≤N。直角顶点是原点。三角形的两条直角边是x轴和y轴。



一天，比塔罗发现他的房间里满是灰尘。开始的时候，房间里有M个灰尘。第i个灰尘（1≤i≤M）位于点（Xi，Yi）。同一点上可能有多个灰尘。

现在，比塔罗计划用扫帚打扫房间。我们把扫帚当作房间里的一条线段，我们把线段的长度称为它的**宽度**。由于比塔罗是很有纪律性的，他只能用如下两种方式使用扫帚：

• 比塔罗把扫帚放在房间里，使它的一端位于原点，并且扫帚是平行于y轴的。然后，他将扫帚沿着x轴的正方向尽可能远地水平移动，移动中保持它与y轴平行并且它的一端位于x轴上。如果扫帚有宽度l，位于（x，y）且x<N - l且y≤l的灰尘都将被移动到（N - l，y）（（N - l，y）处可能存在其他的灰尘）。这叫做操作H。

• 比塔罗把扫帚放在房间里，使它的一端位于原点，并且扫帚是平行于x轴的。然后，他将扫帚沿着y轴的正方向尽可能远地垂直移动，移动中保持它与x轴平行并且它的一端位于y轴上。如果扫帚有宽度l，位于（x，y）且x≤ l且y< N-l的灰尘都将被移动到（x，N - l）（（x，N - l）处可能存在其他的灰尘）。这叫做操作V。

在比塔罗的房间中，Q个事件将按次序发生。事件j(1 ≤j≤ Q)是如下的一种：

• 比塔罗计算灰尘Pj的坐标

• 比塔罗使用宽度为Lj的扫帚执行操作H

• 比塔罗使用宽度为Lj的扫帚执行操作V

• （Aj，Bj）上出现一个新的灰尘。如果在这之前总共有c个灰尘，则这个灰尘编号为c+1

写一个程序，给定房间的直角边长，房间中灰尘的坐标，事件的具体信息，要求计算事件中询问的灰尘坐标。

题目输入：

N M Q

X1 Y1

.

.

.

.

XM YM

（Query 1）

.

.

.

（Query Q）

每个事件j(1 ≤j≤ Q)中包含两个或三个被空格隔开的整数，设Tj为为第一个整数，则每行的含义定义如下：

• 如果Tj=1，该行包含两个整数Tj、Pj，表示在事件j中，比塔罗计算灰尘Pj的坐标。

• 如果Tj=2，该行包含两个整数Tj、Lj，表示在事件j中，比塔罗使用宽度为Lj的扫帚执行操作H

• 如果Tj=3，该行包含两个整数Tj、Lj，表示在事件j中，比塔罗使用宽度为Lj的扫帚执行操作V

• 如果Tj=4，该行包含三个整数Tj、Aj、Bj，表示在事件j中，（Aj，Bj）上出现一个新的灰尘。

保证所有数据为整数。

题目输出：

对于每一个Tj=1的事件，输出一行表示Pj当时的坐标x、y

数据限制：

• 1 ≤ N ≤ 1 000 000 000.

• 1 ≤ M ≤ 500 000.

• 1 ≤ Q ≤ 1 000 000.

• 0 ≤ Xi ≤ N (1 ≤ i ≤ M).

• 0 ≤ Yi ≤ N (1 ≤ i ≤ M).

• Xi + Yi ≤ N (1 ≤ i ≤ M).

• 1 ≤ Pj ≤ (the number of dusts when the event j happens) (1 ≤ j ≤ Q).

• 0 ≤ Lj ≤ N-1 (1 ≤ j ≤ Q).

• 0 ≤ Aj ≤ N (1 ≤ j ≤ Q).

• 0 ≤ Bj ≤ N (1 ≤ j ≤ Q).

• Aj + Bj ≤ N (1 ≤ j ≤ Q).

• There exists at least one event with T j = 1 (1 ≤ j ≤ Q).

子任务：

1. (1 point) M ≤ 2 000, Q ≤ 5 000.

2. (10 points) T j = 1, 2, 4.

3. (11 points) T j = 1, 2, 3, Xj ≤ Xj+1, Yj ≥ Yj+1 (1 ≤ j ≤ M 1).

4. (53 points) T j = 1, 2, 3.

5. (25 points) No additional constraints.

样例一：

输入：

6 2 10

1 1

4 0

4 2 3

3 3

1 1

4 1 2

2 3

2 0

1 4

3 2

1 3

1 2

输出：

1 3

3 2

3 3

6 0

解释：

• 最开始，第一个灰尘位于 (1, 1)，第二个灰尘位于 (4, 0). 图 1 描述了此时的房间.

• 事件1中, 第3个灰尘被加在 (2, 3). 见图2.

• 事件2中, 比塔罗使用宽度为3的扫帚执行操作 V. 于是第一个灰尘被移动至 (1, 3). 见图三.

• 事件3中, 比塔罗计算出第一个灰尘的坐标 (1, 3) .

• 事件 4中, 第4个灰尘被加在(1, 2). 见图4.

• 事件 5中, 比塔罗使用宽度为3的扫帚执行操作 H. 第一个灰尘被移动至(3, 3), 第3个灰尘移动至 (3, 3), 第4个灰尘移动至(3, 2). 见图5

• 事件6中, 比塔罗使用宽度为 0的扫帚执行操作H. 第二个灰尘移动至 (6, 0). 见图6

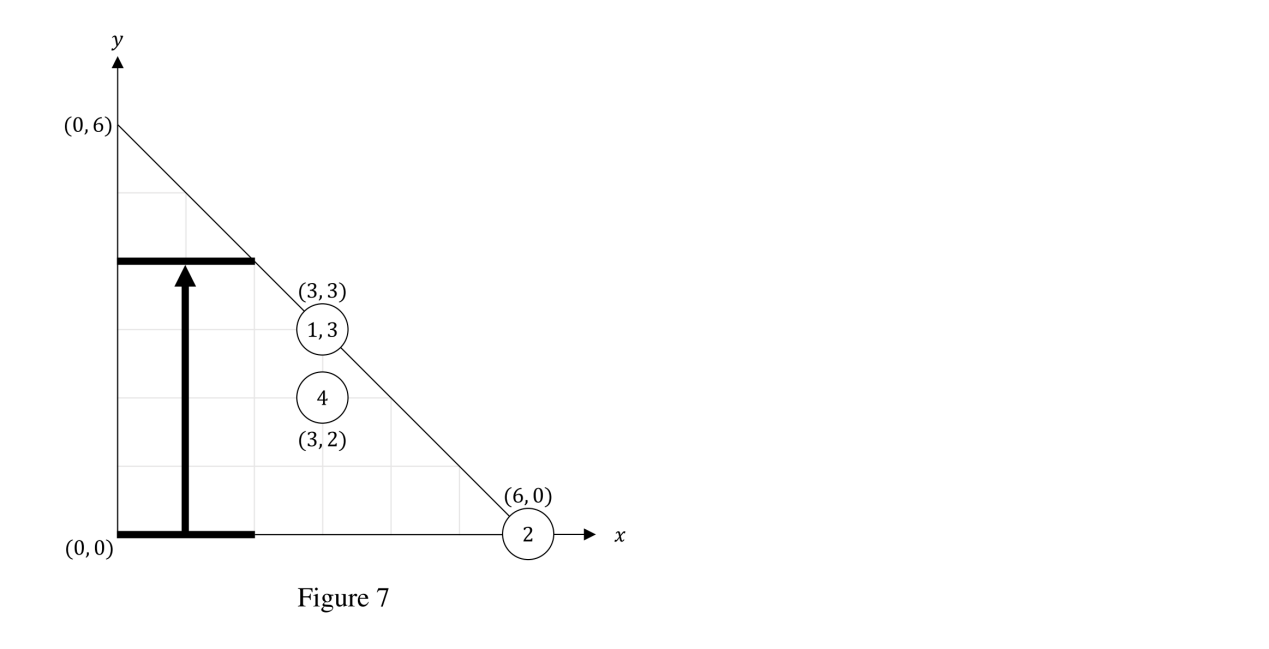
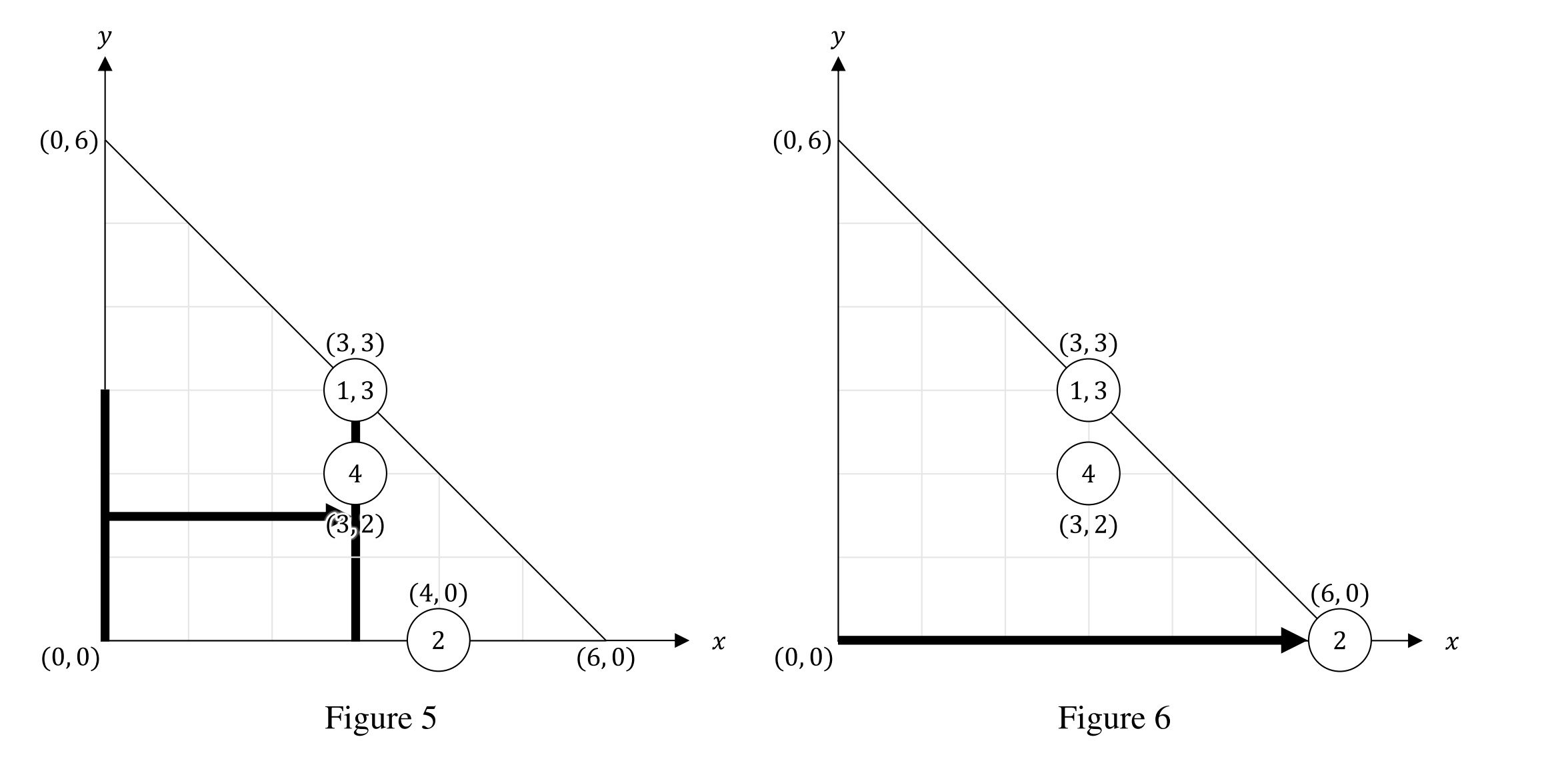
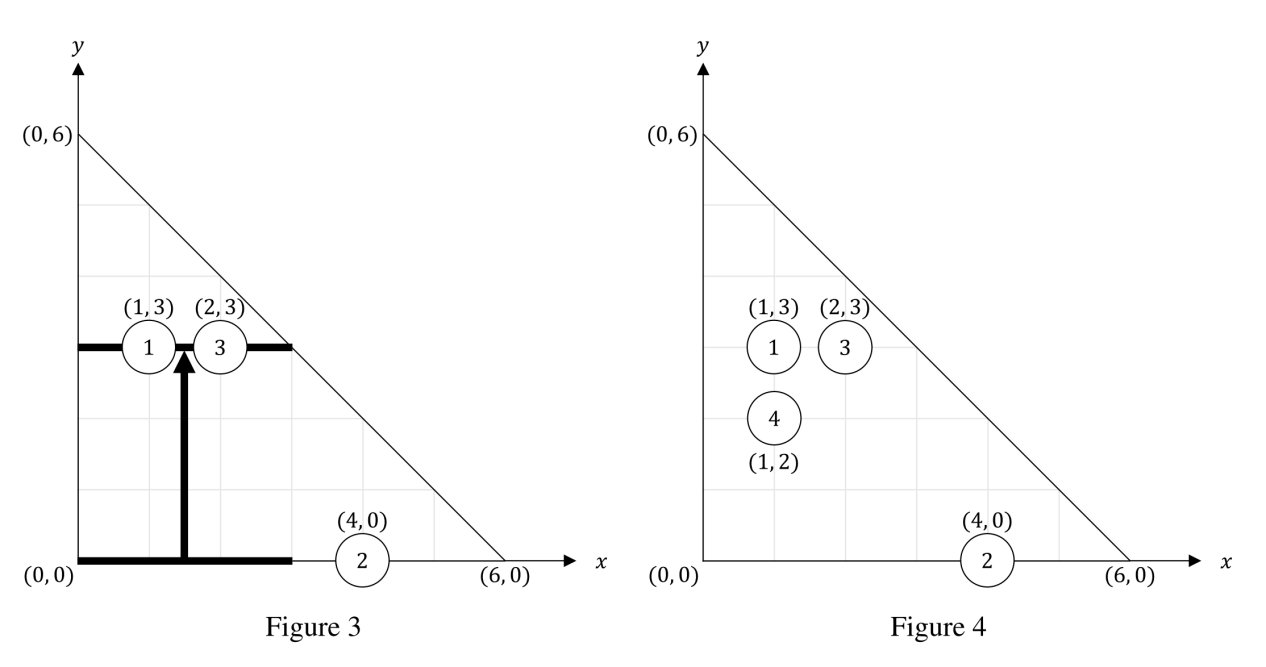
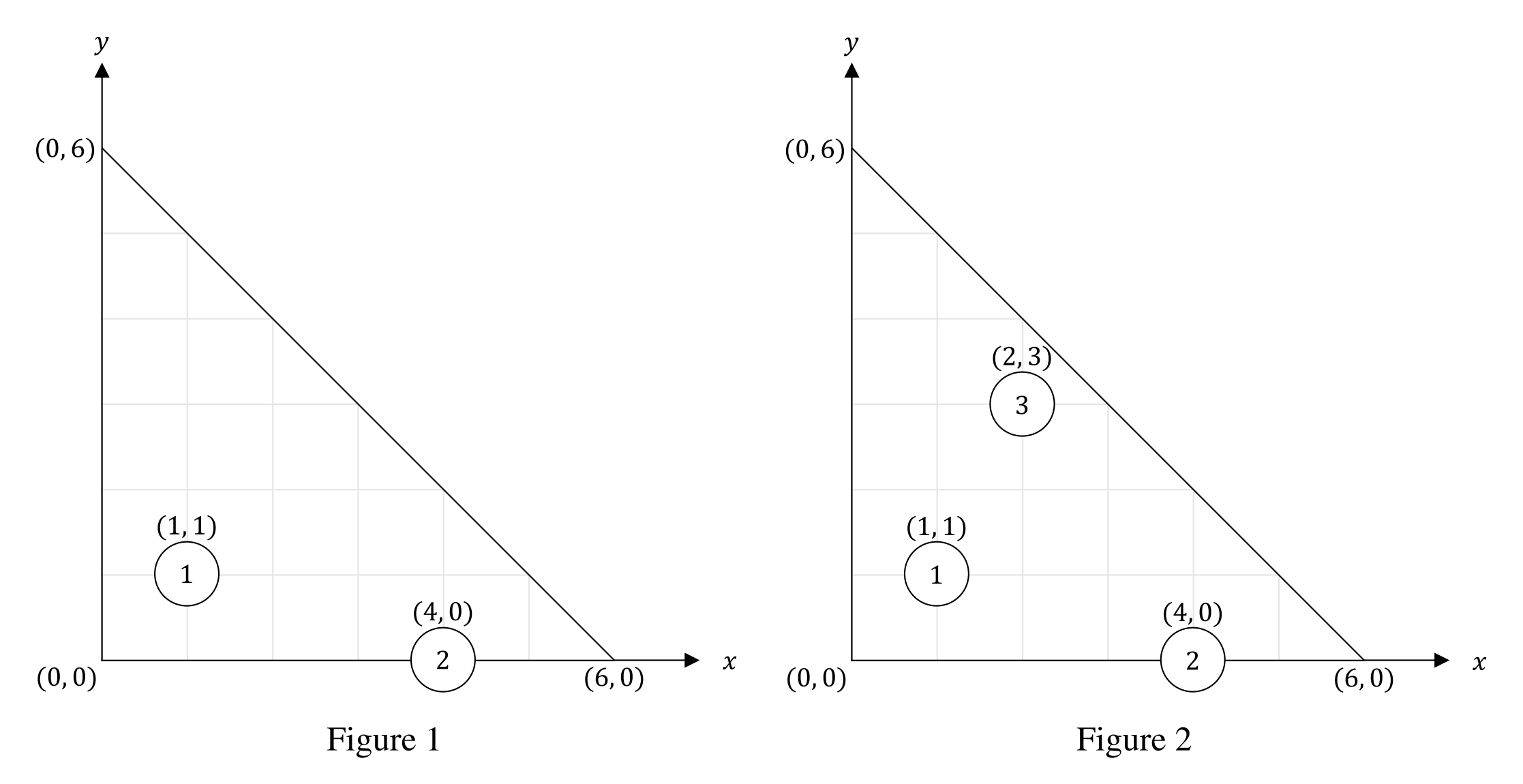
• 事件7中, 比塔罗计算第4个灰尘的坐标(3, 2) .

• 事件8中, 比塔罗使用宽度为 2 的扫帚执行操作 V. 没有灰尘被移动。见图7

• 事件9中, 比塔罗计算第3个灰尘的坐标 (3, 3).

• 事件10中, 比塔罗计算第2个灰尘的坐标(6, 0)

这个样例满足子任务1、5的限制。



样例二：

输入：

9 4 8

2 3

3 1

1 6

4 3

2 6

1 3

2 2

1 4

2 3

1 2

2 4

1 1

输出：

3 6

4 3

7 1

6 3

解释：

这个样例满足子任务1、2、4、5的限制

样例三：

输入：

8 1 8

1 5

4 4 1

2 6

1 2

2 3

4 2 2

2 5

1 1

1 3

输出：

4 1

3 5

3 2

解释：

这个样例满足子任务1、2、5的限制

样例四：

输入：

7 4 9

1 5

2 2

4 2

5 0

2 6

2 3

1 2

3 6

1 4

3 1

1 1

2 2

1 3

输出：

4 2

5 1

1 6

5 2

解释：

这个样例满足子任务1、3、4、5的限制

样例五：

输入：

20 5 25

10 6

0 4

2 1

1 0

2 3

2 18

3 9

4 1 5

4 0 2

3 10

4 3 3

3 3

2 9

4 9 1

3 12

1 4

3 19

1 3

1 9

2 1

1 7

1 6

4 3 3

1 10

1 1

1 5

2 0

1 2

2 2

1 7

输出：

2 17

2 17

9 8

0 17

1 17

3 3

10 10

2 17

2 17

0 17

解释：

这个样例满足子任务1、5的限制