题目描述

AI识别到面板上有N（1 ≤ N ≤ 100）个指示灯，灯大小一样，任意两个之间无重叠。

由于AI识别误差，每次别到的指示灯位置可能有差异，以4个坐标值描述AI识别的指示灯的大小和位置(左上角x1,y1，右下角x2,y2)，

请输出先行后列排序的指示灯的编号，排序规则：

1. 每次在尚未排序的灯中挑选最高的灯作为的基准灯，
2. 找出和基准灯属于同一行所有的灯进行排序。两个灯高低偏差不超过灯半径算同一行（即两个灯坐标的差 ≤ 灯高度的一半）。

输入描述

第一行为N，表示灯的个数
接下来N行，每行为1个灯的坐标信息，格式为：

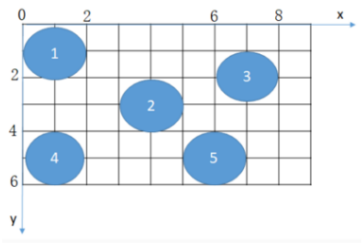
> 编号 x1 y1 2 y2

- 编号全局唯一
- 1 ≤ 编号 ≤ 100
- 0 ≤ x1 < x2 ≤ 1000
- 0 ≤ y1 < y2 ≤ 1000

输出描述

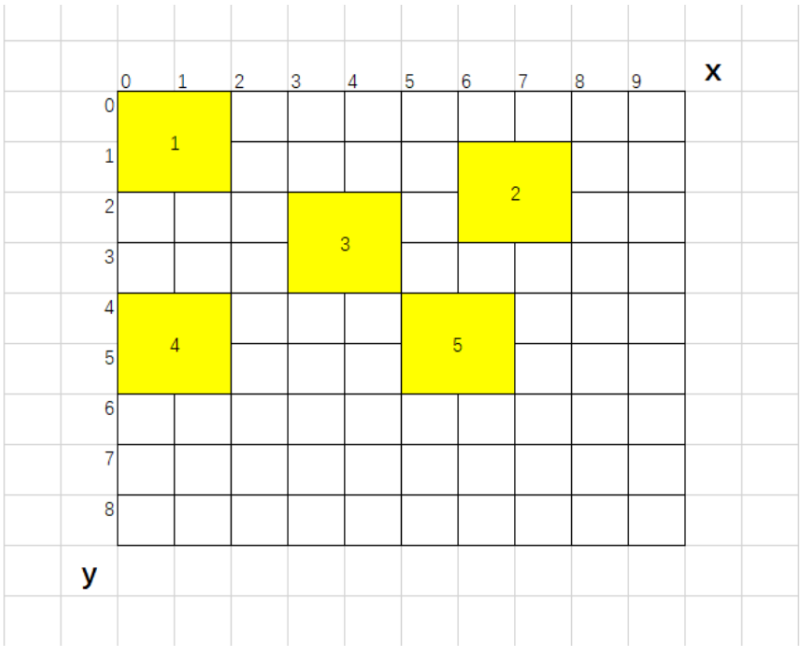
排序后的编号列表，编号之间以空格分隔

用例

输入	5 1 0 0 2 2 2 6 1 8 3 3 3 2 5 4 5 5 4 7 6 4 0 4 2 6
输出	1 2 3 4 5
说明	

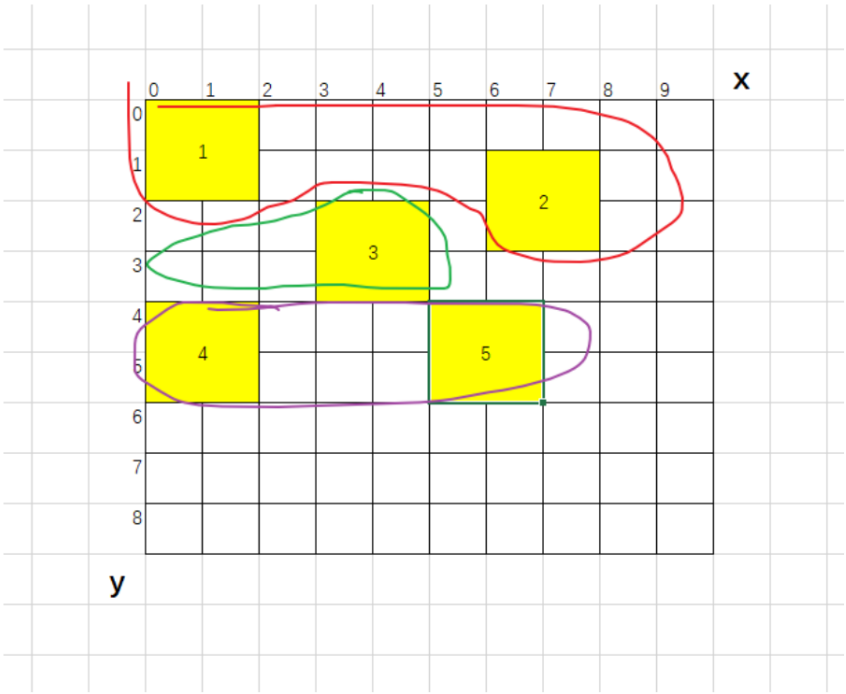
题目解析

题目给的图好像有问题，我画出来是



按照题目意思，每次取出一个最高的灯（y坐标最小的），然后找出和最高灯坐标相差小于等于灯半径的，作为同一行，然后按照x轴坐标进行升序

即存在如下用圈框住的三行，因此打印顺序是：1，2，3，4，5




```java
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
import java.util.StringJoiner;

public class Main {
```

```java
public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);

  int n = sc.nextInt();

  Light[] lights = new Light[n];
  for (int i = 0; i < n; i++) {
    int id = sc.nextInt();
    int x1 = sc.nextInt();
    int y1 = sc.nextInt();
    int x2 = sc.nextInt();
    int y2 = sc.nextInt();
    lights[i] = new Light(id, (x1 + x2) / 2, (y1 + y2) / 2, (x2 - x1) / 2);
  }

  System.out.println(getResult(lights));
}

public static String getResult(Light[] lights) {
  // 按照圆心y坐标升序
  Arrays.sort(lights, (a, b) -> a.y - b.y);

  // ans记录题解
  StringJoiner ans = new StringJoiner(" ");

  // sameRowLights记录同一行的灯
  ArrayList<Light> sameRowLights = new ArrayList<>();
  Light base = lights[0];
  sameRowLights.add(base);

  for (int i = 1; i < lights.length; i++) {
    Light light = lights[i];

    // 如果lights[i]的纵坐标和base的纵坐标相差不超过半径，则视为同一行
    if (light.y - base.y <= base.r) {
      sameRowLights.add(light);
    } else {
      // 否则，不是同一行
      // 针对同一行的灯，再按照横坐标升序
      sameRowLights.sort((a, b) -> a.x - b.x);
```

```
            sameRowLights.forEach(a -> ans.add(a.id + ""));
            sameRowLights.clear();

            // 开始新的一行记录
            base = light;
            sameRowLights.add(base);
          }
        }

        // 注意不要漏了最后一行
        if (sameRowLights.size() > 0) {
          sameRowLights.sort((a, b) -> a.x - b.x);
          sameRowLights.forEach(a -> ans.add(a.id + ""));
        }

        return ans.toString();
    }
}

class Light {
    int id; // 编号
    int x; // 圆心横坐标
    int y; // 圆心纵坐标
    int r; // 圆半径

    public Light(int id, int x, int y, int r) {
        this.id = id;
        this.x = x;
        this.y = y;
        this.r = r;
    }
}
```