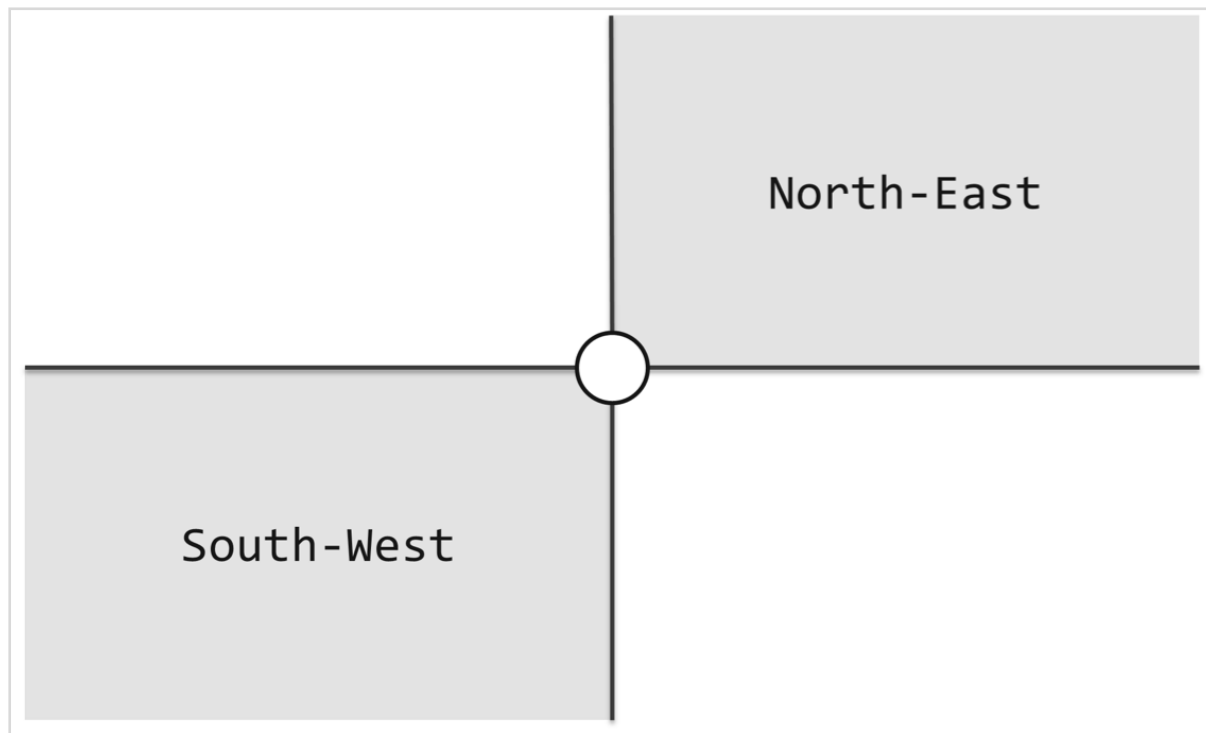


# LightHouse

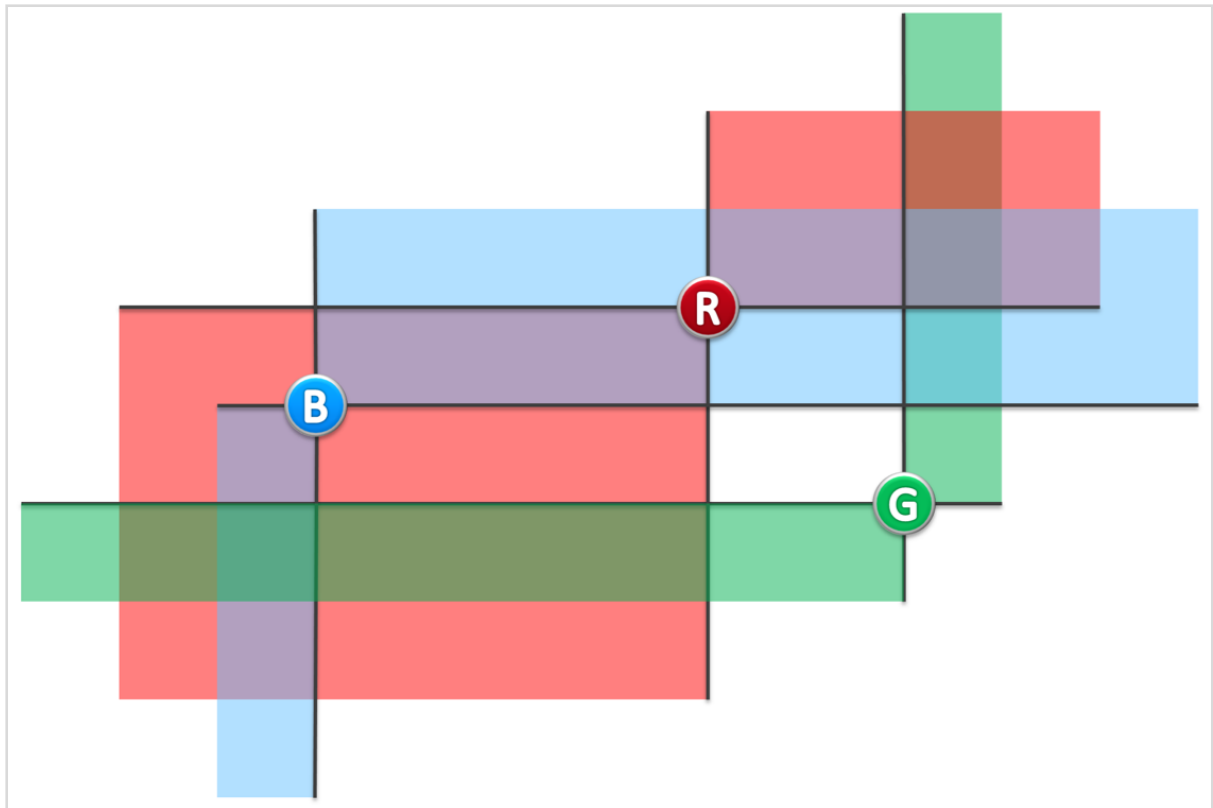
#数据结构邓神

海上有许多灯塔，为过路船只照明。从平面上看，海域范围是 $[1, 10^8] \times [1, 10^8]$ 。



(图一)

如图一所示，每个灯塔都配有一盏探照灯，照亮其东北、西南两个对顶的直角区域。探照灯的功率之大，足以覆盖任何距离。灯塔本身是如此之小，可以假定它们不会彼此遮挡。



(图二)

若灯塔A、B均在对方的照亮范围内，则称它们能够照亮彼此。比如在图二的实例中，蓝、红灯塔可照亮彼此，蓝、绿灯塔则不是，红、绿灯塔也不是。

现在，对于任何一组给定的灯塔，请计算出其中有多少对灯塔能够照亮彼此。

共 $n+1$ 行。

第1行为1个整数 $n$ ，表示灯塔的总数。

第2到 $n+1$ 行每行包含2个整数 $x, y$ ，分别表示各灯塔的横、纵坐标。

对于90%的测例： $1 \leq n \leq 3 \times 10^5$

对于95%的测例： $1 \leq n \leq 10^6$

全部测例： $1 \leq n \leq 4 \times 10^6$

灯塔的坐标 $x, y$ 是整数，且不同灯塔的 $x, y$ 坐标均互异

$1 \leq x, y \leq 10^8$

注意机器中整型变量的范围，C/C++中的int类型通常被编译成32位整数，其范围为 $[-2^{31}, 2^{31} - 1]$ ，不一定足够容纳本题的输出。

时间：2s，内存：256MB

## 如果我们遍历每一元素和其他元素那么必然超时

我们不妨使用归并算法的方式，也就是分而治之

```
void mergeSort(vector v){
    if( ... ) { // 平凡
        return ;
    }
}
```

```
}  
mergeSort(v[0,mi));  
mergeSort(v[mi,hi));  
merge(V_l,V_r);  
}
```

```
int invInside(vector v){  
    if( ... ) { // 平凡  
        return ;  
    }  
    invInside(v[0,mi));  
    invInside(v[mi,hi));  
    invBetween(V_l,V_r);  
}
```