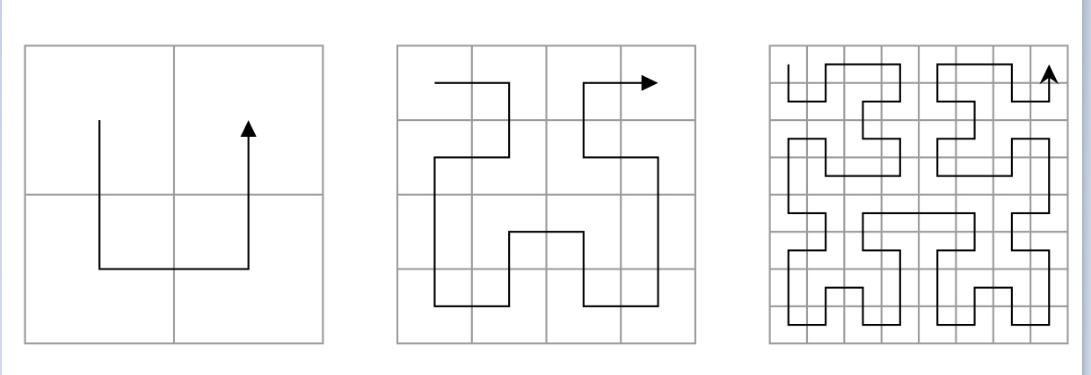
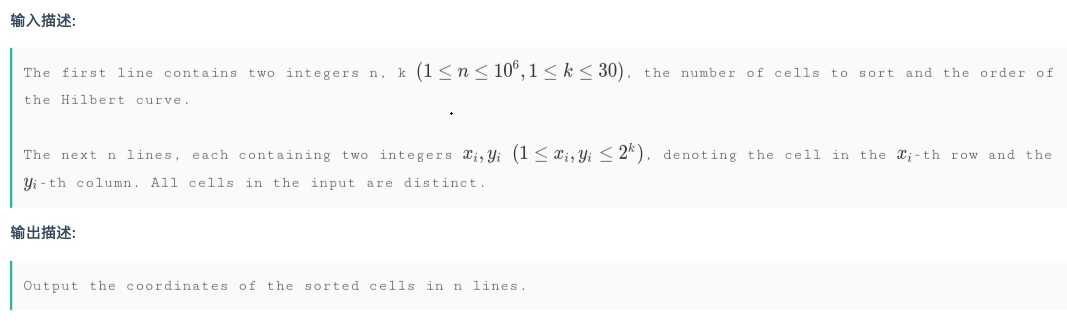
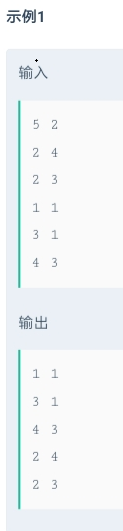
2019牛客第十场 E Helbert Sort （分形/递归）

题意：

给你这样的分形，给你n个点的x,y值，找出他们在第k个分形中的序号（就是按照线的顺序，左上角1,1位置序号为1），并按照序号从小到大输出位置。





分析：

与算法竞赛中的Fractal Streets类似 ，Fractal Streets是根据序号求坐标，而本题是根据坐标求序号。

分形由k->k+1的坐标变换：

左上角：(x,y) -> (y,x)

左下角：(x,y) -> (x+len,y)

右下角：(x,y) -> (x+len,y+len)

右上角：(x,y) -> (len+1-y,-x+2\*len+1)

我们只需要逆着计算这些坐标就可以递归求出，该点位于子分形的哪个位置，然后递归时计算一下序号记录排序输出就可以。

代码：

#include <bits/stdc++.h>

**using** **namespace** std;

**typedef** **long** **long** ll;

**const** **int** maxn = 1000005;

**struct** node{

    ll x , y;

    ll val;

}p[maxn];

**bool** cmp(node a , node b){

**return** a.val < b.val;

}

ll calc(**int** k , ll x, ll y){

**if**(k == 0) **return** 1ll;

    ll len = 1ll << (k - 1);

**if**(x <= len && y <= len){ //1

**return** calc(k - 1 , y , x);

    }**else** **if**(x > len && y <= len){ //2

**return** len\*len + calc(k - 1 , x - len , y);

    } **else** **if**(x > len && y > len){ //3

**return** 2ll \* len\*len + calc(k - 1 , x - len , y - len);

    }**else** { //3

**return** 3ll \* len\*len + calc(k - 1 , 2ll \* len + 1ll - y , len + 1ll - x);

    }

}

**int** main(){

**int** n , k;

**scanf**("%d %d",&n, &k);

**for**(**int** i = 1 ; i <= n ; ++ i){

**scanf**("%lld %lld",&p[i].x,&p[i].y);

        p[i].val = calc(k , p[i].x , p[i].y);

    }

    sort(p + 1 , p + 1 + n , cmp);

**for**(**int** i = 1 ; i <= n ; i ++)

**printf**("%lld %lld\n",p[i].x,p[i].y);

**return** 0;

}