

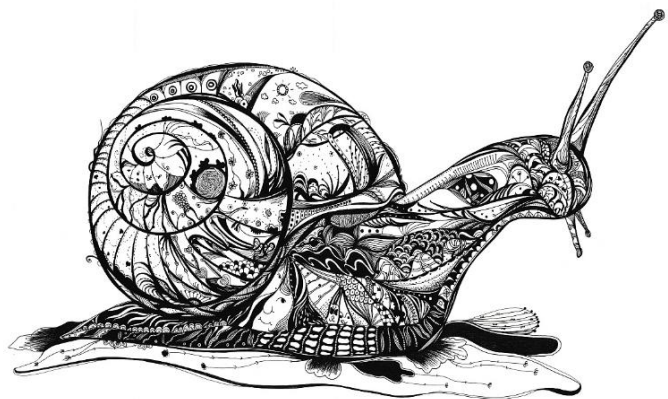
太戈编程  
etiger.vip

# 信奥算法

---

# snail

---



请同学写出题目大意  
已知什么求什么

蜗牛从高度为0开始，每秒高度加a，  
下一秒高度再减b，反复进行，直到高度为h至少几秒？

# 部分分策略讨论

	<b>【数据规模与约定】</b> 测试数据共25组
特殊情况	1号、2号数据： $a=b$
特殊情况	3号、4号数据： $a \geq h$
$O(h)$	5号到20号数据： $h \leq 10^6$
$O(1)$	对于所有数据： $b \leq a \leq 10^{18}$ , $h \leq 10^{18}$

无解情况

每天上升小于等于下落  
 $x \leq y$

并且第一天没有成功  
 $x < h$

特别大

h特别大

x特别大

y特别大

特别小

h特别小

x特别小

y特别小

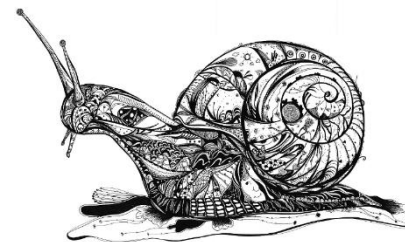
刚刚好

刚好到达

刚好无解

 $x == y$  $x == h$ 

请至少设计3组输入输出  
体现易错点



输入样例

2 2 3

输出样例

forever

输入样例

900 800 1000

输出样例

3

输入样例

11 10 10

输出样例

1

## 变量类型

用int还是long long

```
typedef long long ll;
```

## 无解情况特判

```
7  ll a,b,h;  
8  cin>>a>>b>>h;  
9  ll d=a-b;  
10 if(  
11     cout<<"forever"<<endl;  
12     return 0;  
13 }
```

d表示每2秒  
高度上升几厘米



# 模拟算法

一秒一秒更新高度变化

```
14      ll t=0;  
15      ll pos=0;  
16      while(  
17            
18          if(  
19              pos+=a;  
20          else  
21              pos-=b;  
22      }  
23      cout<<t<<endl;
```

时间复杂度 $O(h)$

# 滚动数组

根据时刻的奇偶性确定上或下

```
14      ll x[2];
15      x[1]=a;
16      x[0]=-b;
17      ll t=0;
18      ll pos=0;
19      while(pos<h){
20          t++;
21          pos+=
22      }
23      cout<<t<<endl;
```

时间复杂度 $O(h)$

# 数学公式

上下上下上下上下上下上下上下上

识别高度变化的规律

上下交错

上升次数=下降次数+1

# 数学公式

上下上下上下上下上下上下上下上

```
9      ll d=a-b;  
14     ll g=max(h-a,0LL);  
15     ll cnt=ceil(g*1.0/d);  
16     ll ans=cnt*2+1;
```

请写出变量的精准含义

d表示每2秒高度上升几厘米

g表示先向上爬a厘米后剩余的距离

cnt表示剩余距离还需要几个2秒的爬行

ans表示总共需要爬几秒

# 太戈编程

3111

# 费用平摊



SHANGHAI



HONGKONG

请同学写出题目大意  
已知什么求什么

已知长度为 $n$ 的整数序列 $p[]$ , 求有多少  
个子段和恰为 $m$ 的倍数?

请同学阅读[数据规模和约定]  
识别部分得分点

对于10%数据,  $m=1$ 。

对于20%数据,  $m \leq 2$ 。

对于30%数据,  $n \leq 100, 0 \leq p[i] \leq 100$

对于50%数据,  $n \leq 1000$ 。

对于100%数据,  $n \leq 200000, m \leq 1000000$ , 保证 $p[i]$ 绝对值均  
不超过10000。



LONDON



PARIS

输入

5 1  
2 3 5 1 5

6 2  
2 4 1 8 3 9

输出?

15

9



# 暴力枚举

```
10 void solveBF(){
11     ll ans=0;
12     for(ll i=1;i<=n;++i){
13         ll sum=0;
14         for(ll j=i;j<=n;++j){
15             sum=(sum+p[j])%m;
16             
17         }
18     }
19     cout<<ans<<endl;
20 }
```

请写出时间复杂度

$O(n^2)$

# 思考如何加速

连续和/子段和



前缀和做差

子段和为 $m$ 倍数



2个前缀和  
模 $m$ 同余

n=5  
m=2

$s[]$  是  $p[]$  的前缀和数组

	i=0	i=1	i=2	i=3	i=4	i=5
$p[i]$	-	2	4	1	8	3
$s[i]$	0	2	6	7	15	18
$s[i] \% m$	0	0	0	1	1	0

对  $s[i]$  按照除以  $m$  的余数进行统计

$cnt[x]$  表示有几个  $s[i]$  除以  $m$  的余数 恰为  $x$

$cnt[0]$  为 4,  $cnt[1]$  为 2

答案 =  $4 * (4 - 1) / 2 + 2 * (2 - 1) / 2 = 7$

## 前缀和+同余系数

```
24 void solve(){
25     for(ll i=1;i<=n;++i){
26         s[i]=s[i-1]+p[i];
27         
28         cnt[s[i]]++;
29     }
30     
31     ll ans=0;
32     for(ll r=0;r<m;++r){
33         if(cnt[r]<=1)continue;
34         ans+=
35     }
36     cout<<ans<<endl;
37 }
```

可能有负数参与取模

$s[0]$ 为0参与计数

能否删除第33行?

请写出时间复杂度

$O(n+m)$

## 前缀和+同余系数

对于10%数据,  $m=1$ 。  
对于20%数据,  $m \leq 2$ 。  
对于30%数据,  $n \leq 100$ ,  $0 \leq p[i] \leq 100$   
对于50%数据,  $n \leq 1000$ 。  
对于100%数据,  $n \leq 200000$ ,  $m \leq 1000000$ , 保证 $p[i]$ 绝对值均不超过10000。

```
const ll N=200009;  
const ll M=1000009;  
  
ll s[N];  
ll cnt[M];
```

请设置N和M的大小

## 大小数据分离 确保部分分

```
38 int main(){
39     freopen("share.in","r",stdin);
40     freopen("share.out","w",stdout);
41     input();
42     if(n<=2000)
43         solveBF();
44     else
45         solve();
46     return 0;
47 }
```

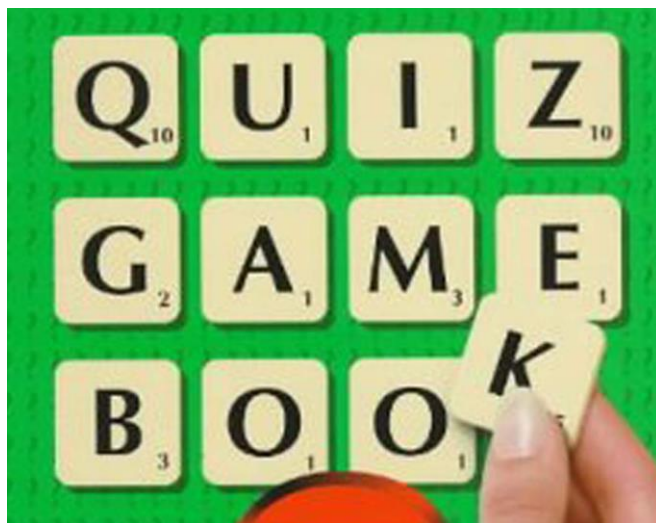


# 太戈编程

2718



# 单词构造



输入

4 5  
qonoc  
ohoho  
hlaqo  
mirko

输出?

qohlaqko

q	o	n	o	c
o	h	o	h	o
h	l	a	q	o
m	i	r	k	o

请同学写出题目大意  
已知什么求什么

$n*m$ 字符,从左上角到右下角走,可以向下或右,取出走过的字符组成字符串,要求字典序最小。

请同学阅读[数据规模和约定]  
识别部分得分点

对于20%数据,  $m=1$ 。

对于30%数据,  $n \leq 100$ ,  $m \leq 100$ 。

对于100%数据,  $n \leq 2000$ ,  $m \leq 2000$ 。

# 观察发现

首个字母一定是左上角

第二个字母可以二选一,挑字典序小的

按照斜线的方向逐个确定答案字符串

平局时如何处理?

枚举每格可走的2种方向: 向右或向下

平局时维护多个最优候选人

暴力  
枚举

贪心  
变种

## 大小数据分离 确保部分分

```
59 int main(){
60     freopen("word.in", "r", stdin);
61     freopen("word.out", "w", stdout);
62     input();
63     if(m==1)
64         print();
65     else if(n+m<=15)
66         solveBF();
67     else
68         solve();
69     return 0;
70 }
```

## 暴力枚举

枚举每格可走的2种方向：向右或向下

对所有可能的字符串进行字典序比较

```
16 string ans,word;
17 void dfs(ll x,ll y){
26 void solveBF(){
27     ans=word="";
28     for(ll i=1;i<=n+m-1;++i){
29         ans+='z';
30         word+=' ';
31     }
32     dfs(1,1);
33     cout<<ans<<endl;
34 }
```

```
16 string ans,word;
17 void dfs(ll x,ll y){
18     
19     if(x==n&&y==m){
20         ans=min(ans,word);
21         return ;
22     }
23     if(x+1<=n) dfs(x+1,y);
24     
25 }
```

请写出时间复杂度

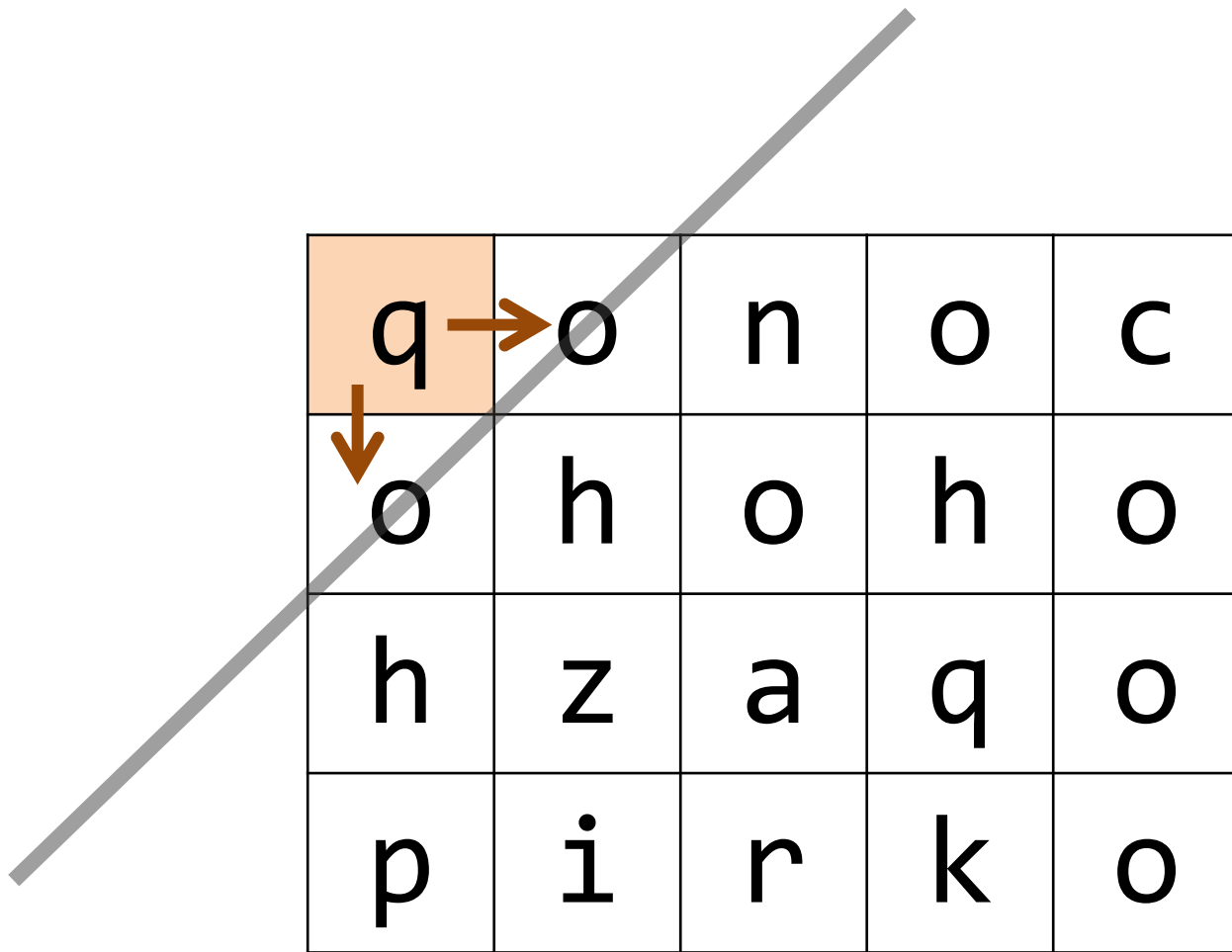
$O(2^{n+m})$



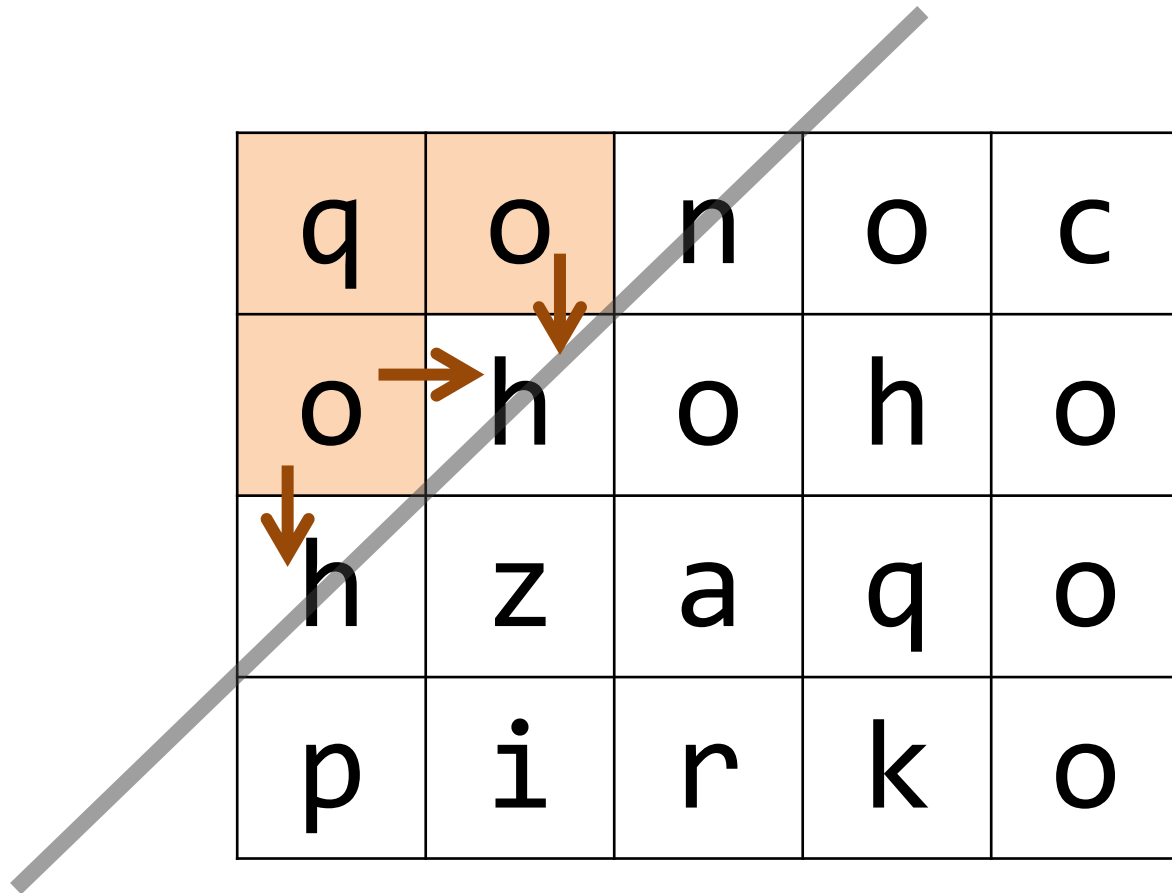
## 贪心 变种

按照斜线的方向逐个确定答案字符串

平局时维护多个最优候选人



答案字符串 = q???????



答案字符串 = qo??????

q	o	n	o	c
o	h	o	h	o
h	z	a	q	o
p	i	r	k	o

答案字符串 = qohoaqko

```
35 bool ok[N][N];
36 void solve(){
37     ans="";
38     ok[1][1]=1;
39     for(ll step=1;step<=n+m-1;++step){
40         char mn='z';
41         for(ll x=1;x<=step&& x<=n;++x){
42             ll y=step+1-x;
43             if(y>m)continue;
44             if(!ok[x][y])continue;
45             mn=min(mn,d[x][y]);
46         }
47         ans+=mn;
48         for(ll x=1;x<=step&& x<=n;++x){
49             ll y=step+1-x;
50             
51             if(!ok[x][y])continue;
52             if(d[x][y]!=mn)continue;
53             if(x+1<=n) ok[x+1][y]=1;
54             
55         }
56     }
57     cout<<ans<<endl;
58 }
```

请写出时间复杂度

 $O(nm)$

## 大小数据分离 确保部分分

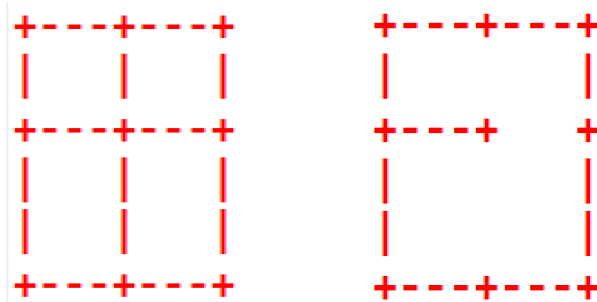
```
59 int main(){
60     freopen("word.in", "r", stdin);
61     freopen("word.out", "w", stdout);
62     input();
63     if(m==1)
64         print();
65     else if(n+m<=15)
66         solveBF();
67     else
68         solve();
69     return 0;
70 }
```



# 太戈编程

2720

draw





请同学写出题目大意  
已知什么求什么

平面上有一个长方形，  
给定 $n$ 条纵向分割线和 $m$ 条横向分割线，  
得到 $(n+1)(m+1)$ 个平面方格，  
要删掉一些小线段使方格连通，  
求删掉线段的最短总长度。

## 理解核心操作

每次删除小线段  
需要连接两个原本不连通的2个区域

原本连通的2个区域之间  
不需要删除分割线段

请同学阅读[数据规模和约定]  
识别部分得分点

**【数据规模与约定】**

1号数据：保证 $1 \leq A, B \leq 20$ ， $0 \leq n, m \leq 10$

2号数据：保证 $0 \leq n, m \leq 100$

3号数据：保证 $m = 1$

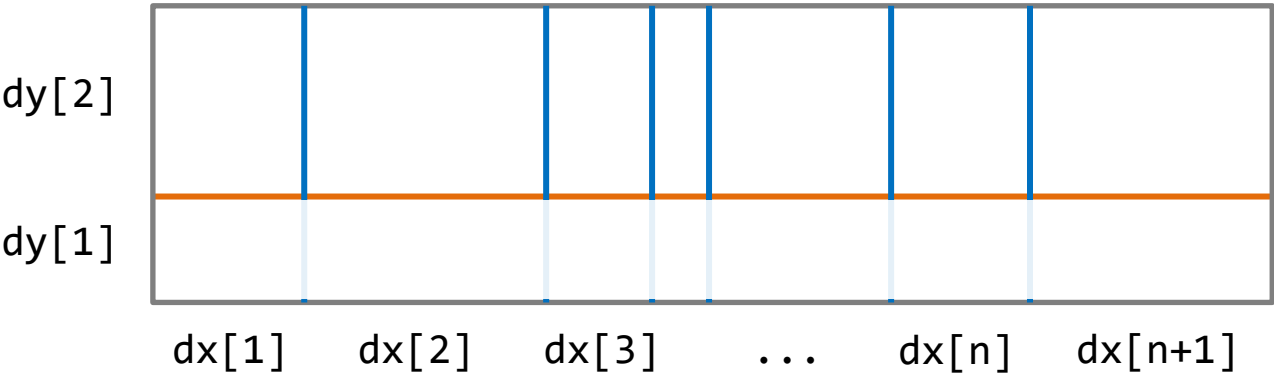
对于所有数据：保证 $1 \leq A, B \leq 1000000000$ ，  
 $0 \leq n, m \leq 25000$

简化问题  
可能启发灵感



```
48 int main(){
49     freopen("draw.in", "r", stdin);
50     freopen("draw.out", "w", stdout);
51     input();
52     if(m==1)
53         solveM1();
54     else
55         solve();
56     return 0;
57 }
```

m=1: 恰有1条横向分割线



纵向小线段长度	$n$ 条 $dy[1]$ , $n$ 条 $dy[2]$	排序: $dy[1] \leq dy[2]$
横向小线段长度	$dx[1], dx[2], \dots, dx[n+1]$	排序: 从小到大

格子总数	$2(n+1)$
原始线段数	$3n+1$
要删除的线段数	格子总数-1 = $2(n+1) - 1 = 2n+1$

重大发现	打通左右通道: $n$ 条长度 $dy[1]$ 的小线段可以先删除 $n$ 条长度 $dy[2]$ 的小线段 和 $n+1$ 条横向小线段里挑最短的 $n+1$ 条删除
------	---

```

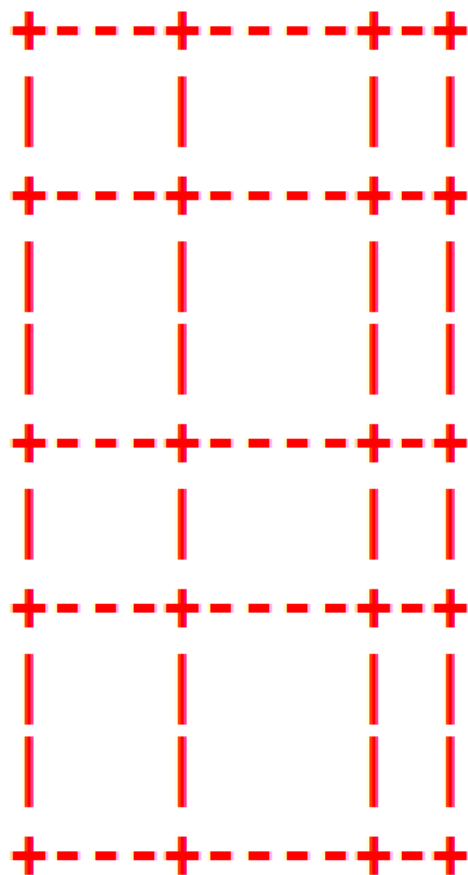
12 void solveM1(){
13     dy[1]=y[1];
14     dy[2]=b-y[1];
15     if(dy[1]>dy[2]) swap(dy[1],dy[2]);
16     ll ans=dy[1]*n;
17     x[n+1]=a;
18     sort(x+1,x+n+2);
19     for(ll i=1;i<=n+1;++i) dx[i]=x[i]-x[i-1];
20     for(ll i=1;i<=n;++i) 
21 
22     for(ll i=1;i<=n+1;++i) ans+=dx[i];
23     cout<<ans<<endl;
24 }

```

时间复杂度 $O(n)$

总结 $m=1$ 情况的算法

推广到满分算法

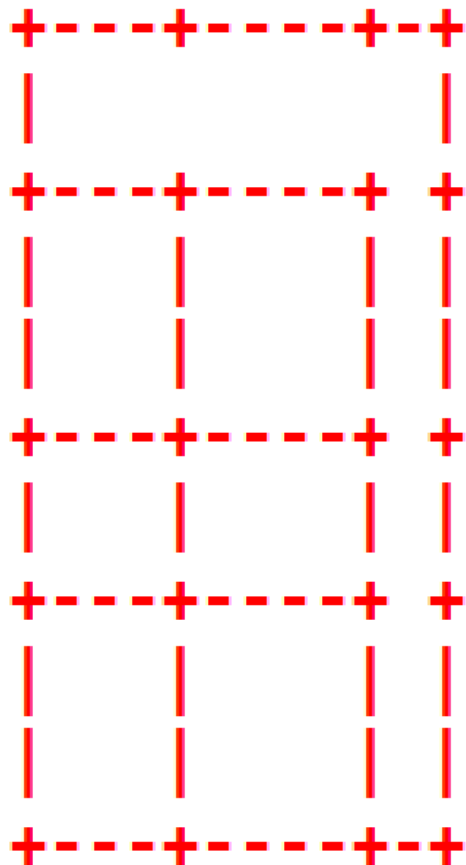


$n=2, m=3$

打通左右通道

打通上下通道





两个方向分别  
贪心挑最短线段

打通左右通道

删除2段纵向小线段

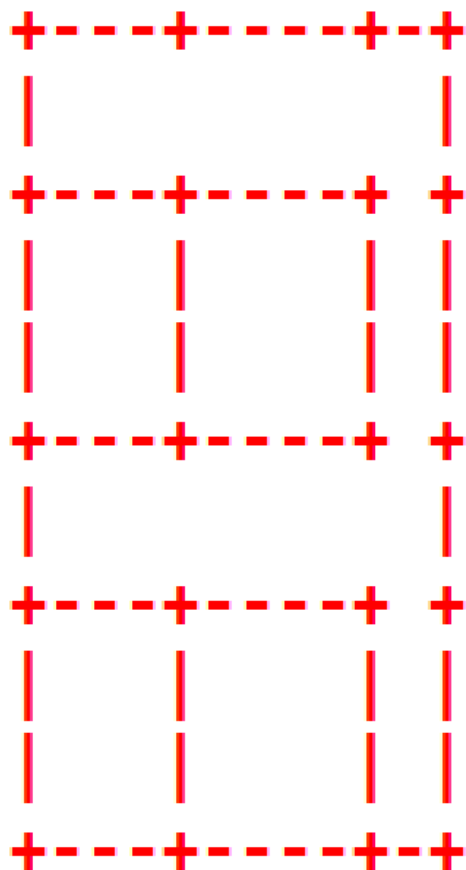
打通上下通道

删除3段横向小线段

完成这批操作后

横向看：存在3个不连通的空间

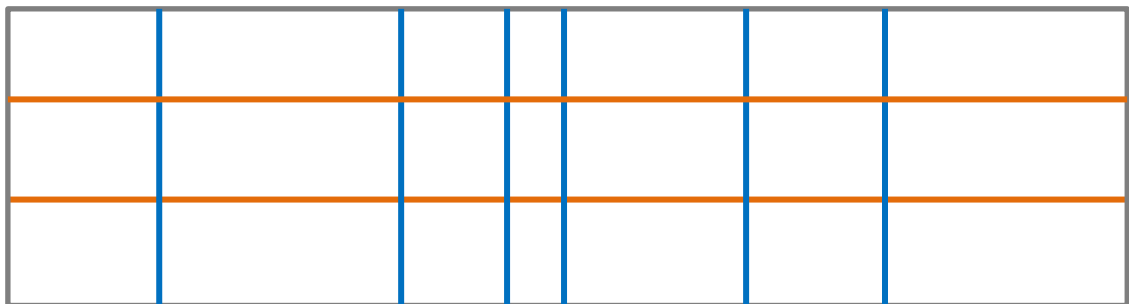
纵向看：存在4个不连通的空间



继续贪心挑最短线段

批量删除2段纵向线段

完成这批操作后  
横向看：存在3个不连通的空间  
纵向看：存在3个不连通的空间



格子总数	$2(n+1)$
原始线段数	$3n+1$
要删除的线段数	格子总数-1 = $2(n+1) - 1 = 2n+1$

重大发现	打通左右通道：n条长度dy[1]的小线段可以先删除 n条长度dy[2]的小线段 和 n+1条横向小线段里 挑最短的n+1条删除
------	---

原始格子数量

$$(n+1)(m+1)$$

原始小线段数量

$$n*(m+1)+m*(n+1)$$

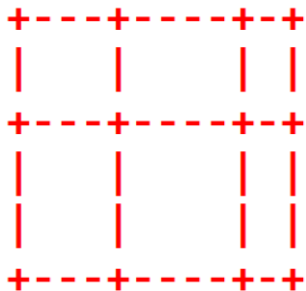
举例： $n=2, m=1$ ，共 $3*2$ 格，7边

要删除线段的数量

$$\text{原始格子数} - 1 = (n+1)(m+1) - 1$$

留下几条？

$$\begin{aligned} \text{原始小线段数量} - \text{删除线段数量} \\ = n*m \end{aligned}$$



# 图论建模

每个节点对应什么？

格子

每条边对应什么？

格子间的隔板

联想：最小生成树

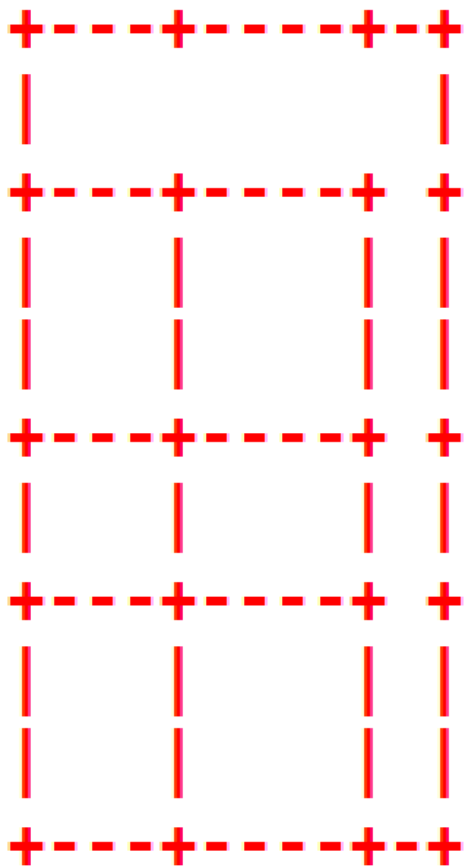
点数太多，边数太多

Kruskal算法(贪心思路)

批量选边

$n=2$  纵向分割线数量

$m=3$  横向分割线数量



$cX$ 表示已经批量删除过几批横向线段

$cY$ 表示已经批量删除过几批纵向线段

$cX=1$

$cY=1$

完成这批操作后

横向看：存在 $n+1$ 个不连通的空间

纵向看：存在 $m+1$ 个不连通的空间

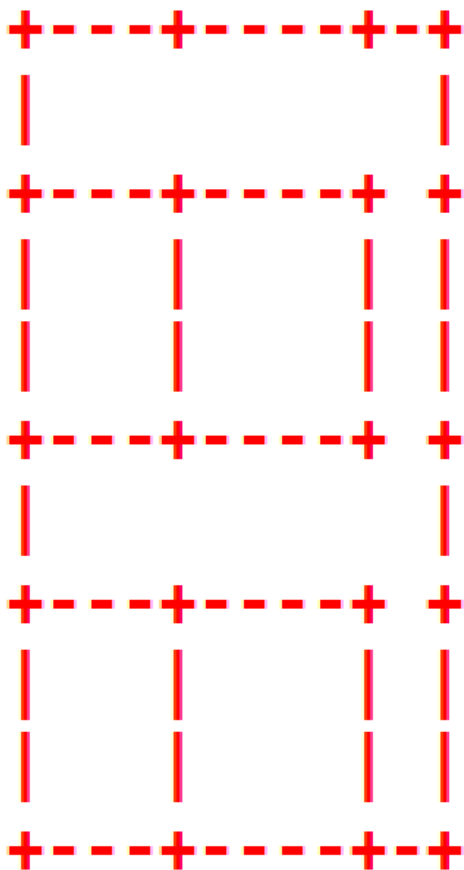
完成这批操作后

横向看：存在 $n+2-cX$ 个不连通的空间

纵向看：存在 $m+2-cY$ 个不连通的空间

$n=2$  纵向分割线数量

$m=3$  横向分割线数量



$cX$ 表示已经批量删除过几批横向线段

$cY$ 表示已经批量删除过几批纵向线段

$cX=1$

$cY=2$

完成这批操作后

横向看：存在 $n+2-cX$ 个不连通的空间

纵向看：存在 $m+2-cY$ 个不连通的空间

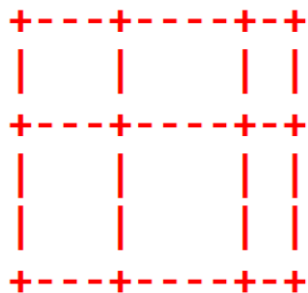
```

26 x[n+1]=a;
27 y[m+1]=b;
28 sort(x+1,x+n+2);
29 sort(y+1,y+m+2);
30 for(ll i=1;i<=n+1;++i) dx[i]=x[i]-x[i-1];
31 for(ll i=1;i<=m+1;++i) dy[i]=y[i]-y[i-1];
32 sort(dx+1,dx+n+2);
33 sort(dy+1,dy+m+2);

```

dx[] 储存横向小线段长度

dy[] 储存纵向小线段长度

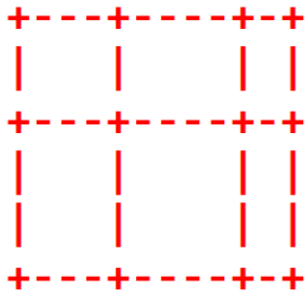




```

34  ll ans=dx[1]*m+dy[1]*n;
35  ll cX=1,cY=1;
36  for(ll i=1;i<=n+m;++i){
37      if(cX+1<=n+1&&dx[cX+1]<dy[cY+1]){
38          ++cX;
39          ans+=dx[cX]*(m+1-cY);
40      }
41      else{
42          ++cY;
43          ans+=dy[cY]*(n+1-cX);
44      }
45  }

```



横向看：存在 $n+2-cX$ 个不连通的空间  
 纵向看：存在 $m+2-cY$ 个不连通的空间

# 太戈编程

2775