

太戈编程
etiger.vip

信奥算法

阶乘差

请同学简述题意
突出核心要点

已知 a, b
求 $(b! - a!)$ 模 10000000009 取余数

求阶乘

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int n;
5      cin>>n;
6      long long sum=1; //long long 只能存20!
7      //int sum=1; //int只能存12!
8      for(int i=1; i<=n; i++) {
9          sum=sum*i;
10         cout<<i<<"!="<<sum<<endl;
11     }
12     return 0;
13 }
```

错误代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4      freopen("factorial.in", "r", stdin);
5      freopen("factorial.out", "w", stdout);
6      int a,b,m;
7      cin>>a>>b>>m;
8      long long sum1=1, sum2=1;
9      for(int i=1; i<=a; i++)
10         sum1=sum1*i;
11     for(int i=1; i<=b; i++)
12         sum2=sum2*i;
13     cout<<(sum1-sum2)%m<<endl;
14     return 0;
15 }
```

当a, b超过20则溢出

算法步骤

用常数MOD记录模数

```
3 typedef long long ll;  
4 const ll MOD=1000000009;
```

求 $a!$ 模MOD取余数,记作f1

求 $b!$ 模MOD取余数,记作f2

```
10 ll f1=1,f2=1;  
11 for(int i=2;i<=a;i++) f1=f1*i%MOD;  
12 for(int i=2;i<=b;i++) f2=f2*i%MOD;
```

每一步计算都需要对中间结果取模

易错点：中间结果太大可能溢出

算法步骤

用常数MOD记录模数

```
3 typedef long long ll;  
4 const ll MOD=1000000009;
```

求 $a!$ 模MOD取余数,记作f1

求 $b!$ 模MOD取余数,记作f2

```
10 ll f1=1,f2=1;  
11 for(int i=2;i<=a;i++) f1=f1*i%MOD;  
12 for(int i=2;i<=b;i++) f2=f2*i%MOD;
```

答案是 $f2-f1$ 吗?

答案是 $(f2-f1)\% \text{MOD}$ 吗?

算法步骤

用常数MOD记录模数

```
3 typedef long long ll;  
4 const ll MOD=1000000009;
```

求a! 模MOD取余数, 记作f1

求b! 模MOD取余数, 记作f2

```
10 ll f1=1, f2=1;  
11 for(int i=2; i<=a; i++) f1=f1*i%MOD;  
12 for(int i=2; i<=b; i++) f2=f2*i%MOD;  
  
13 ll ans=(f2-f1+MOD)%MOD;  
14 cout<<ans<<endl;
```

易错点：负数取模还是负数, 需要加到正数

代码2

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  const ll MOD=1000000009;
5  ll a,b,ans,f1,f2;
6  int main(){
7      freopen("factorial.in","r",stdin);
8      freopen("factorial.out","w",stdout);
9      cin>>a>>b;
10     ll f1=1,f2=1;
11     for(int i=2;i<=a;i++) f1=f1*i%MOD;
12     for(int i=2;i<=b;i++) f2=f2*i%MOD;
13     ll ans=(f2-f1+MOD)%MOD;
14     cout<<ans<<endl;
15     return 0;
16 }
```

太戈编程

1800

文武双全

请同学写出题目大意

输入

8 5 3

2 4 7 5 2 5 7 7

3 1 2 2 2 4 2 4

输入

8 5 3

2 4 7 5 2 5 7 7

3 1 2 2 2 4 2 4

输出

39

请同学设计部分分策略

【数据规模与约定】

1号数据: $n \leq 10$

2号、3号、4号数据: $n \leq 20$

5号数据: 所有 $x[i]$ 均为1, 所有 $y[i]$ 均为2

对于所有数据: $n \leq 100000$

模型要素

集合最优划分

必须二选一

最大化总的分组打分

$$\max\{ \text{sum}_X\{x_i\} + \text{sum}_Y\{y_i\} \}$$

算法分析

尝试贪心法

错误贪心

每次选唱歌最高分,加入合唱团

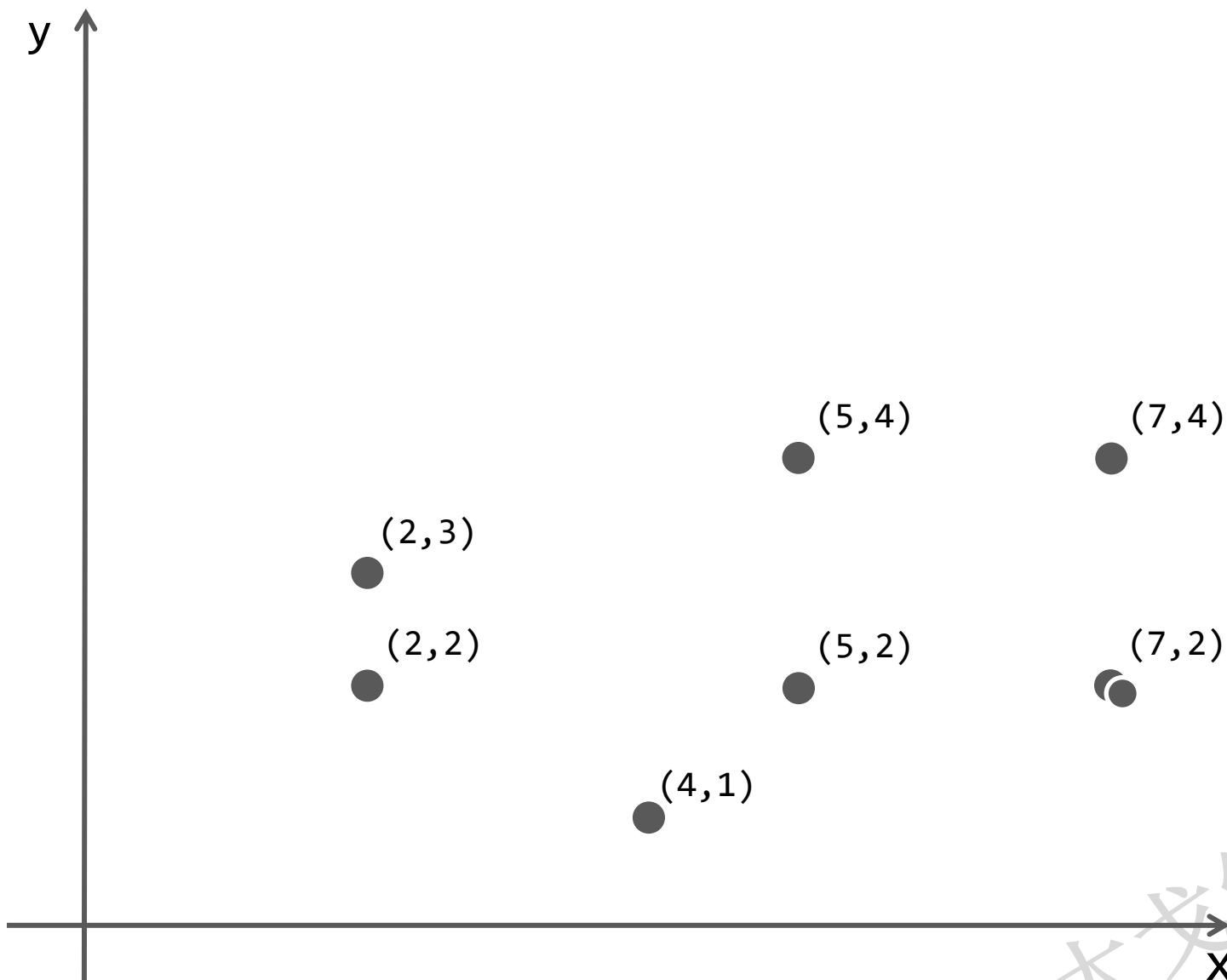
每次选长跑最高分,加入长跑队

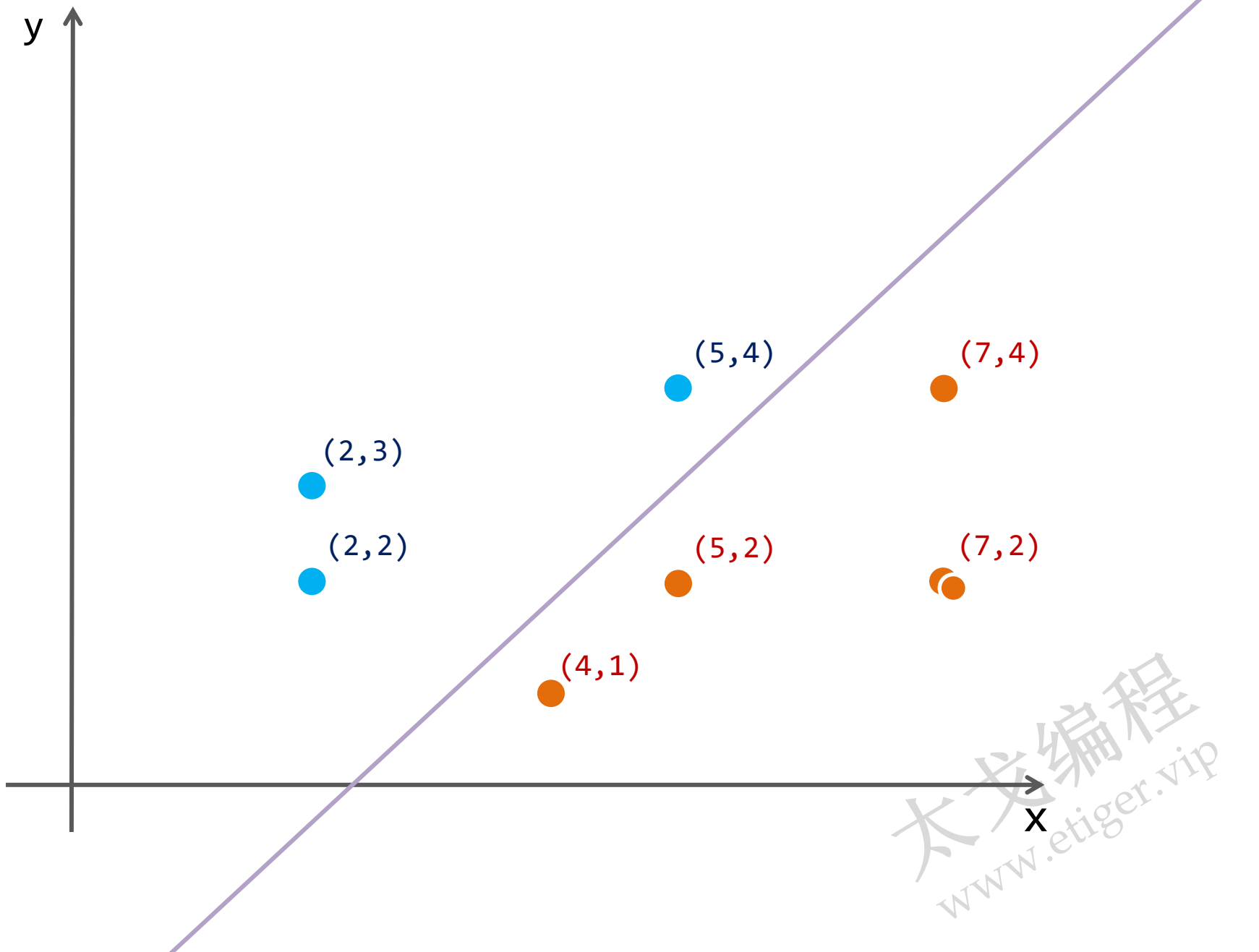
错误原因

只考虑了收益
忽略了机会成本

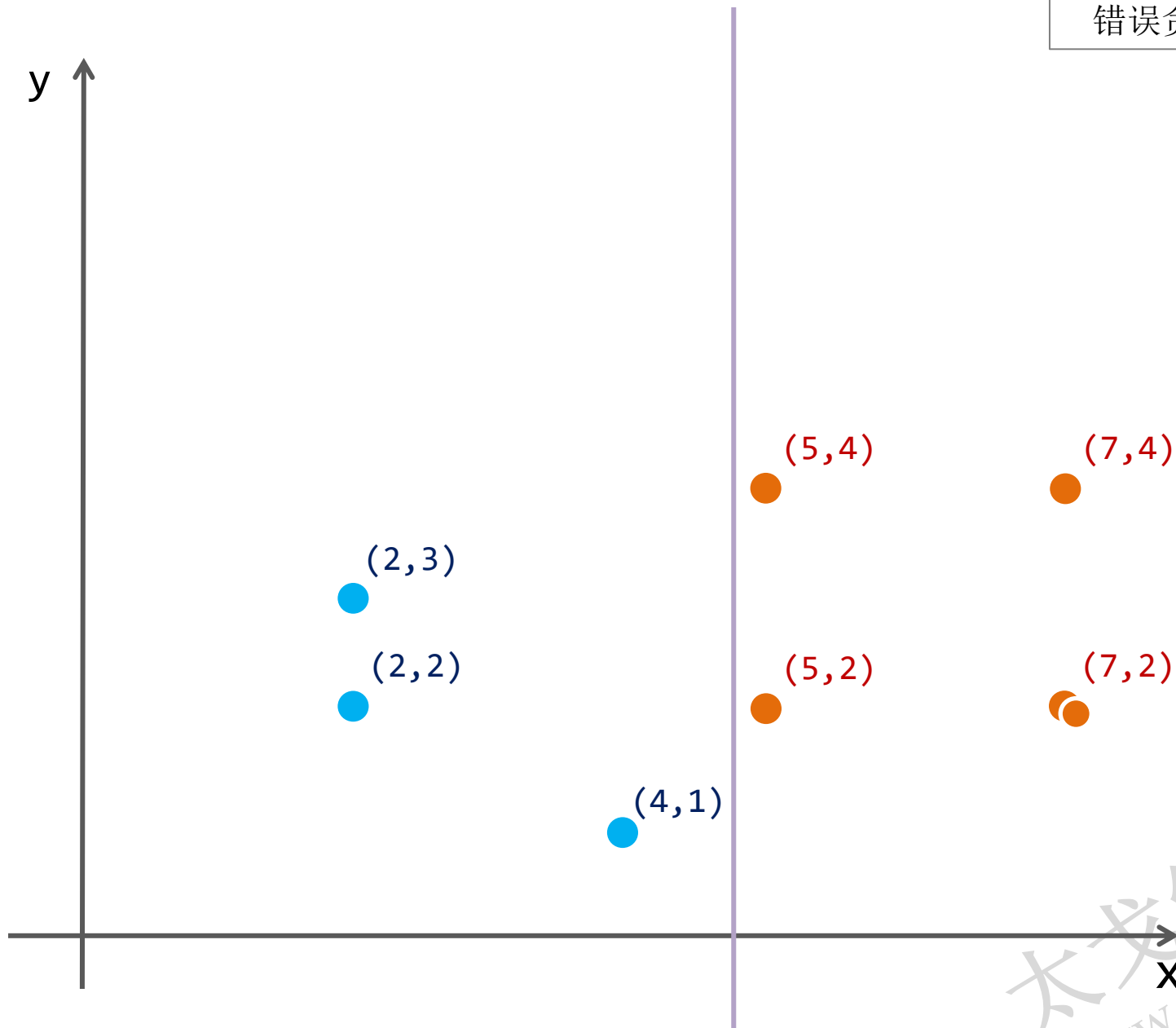
启发思路

每个元素有两个属性
二维平面点可视化





错误贪心的演示



算法分析

正解贪心法

相对收益	$d[i] = x[i] - y[i]$
	每次选相对收益最高的 加入相应社团

按照 $d[]$ 从大到小排序 前 n_X 个加入合唱团 后 n_Y 个加入长跑队

如何证明正确性?
反证+调整

算法分析

正解贪心法

相对收益

$$d[i] = x[i] - y[i]$$

每次选相对收益最高的
加入相应社团

等效操作

先将所有人放入Y组
然后贪心地选出人员改成X组

```
13 struct Info{
14     int v,id;
15     bool operator<(const Info&x) const{
16         return v>x.v || v==x.v&&id>x.id;
17     }
18 };
19 Info d[N];
```

d[i]需要包含两个信息
相对收益,原始编号


```
19 Info d[N];
20 void solve(){
21     for(int i=1;i<=n;++i)
22         d[i]=
23
24     int ans=0;
25
26
27
28
29     printf("%d\n",ans);
30 }
```

太戈编程

3075

做题机器

请同学写出题目大意
已知什么求什么

L条做题记录:非负 x_i 表示写了几行,负数 x_i 表示删除 $|x_i|$ 行。存在未知数 n ,若写的行数达到 n 就算完成1题。恰好完成 k 题,求 n 的最小值和最大值。

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】

对于20%的数据, $L \leq 10$

对于40%的数据, $L \leq 100$

对于60%的数据, $L \leq 2000$

对于100%的数据, $L \leq 100000$, 保证 $-10^{18} \leq n \leq 10^{18}$,
 $-10^9 \leq x_i \leq 10^9$

枚举答案

确认问题的单调性

较小 n 对应完成的题数
大于等于
较大 n 对应完成的题数

二分答案

最大化 n ，满足：完成题数 $\geq k$
最小化 n ，满足：完成题数 $\leq k$

$AC(n)$ 表示给定 n 时的完成题数

二分答案

最大化 n : 使完成题数 $\geq k$

是否可能无解?

最小化 n : 使完成题数 $\leq k$

是否可能无解?

```
29     ll l=1;
30     ll r=1e18;
31     ll mx=0;
32     while(l<=r){
33         ll mid=l+(r-l)/2;
34         if(AC(mid)>=k) mx=mid,l=mid+1;
35         else r=mid-1;
36     }
37
38
39
40
41
42
43
44
45     if(AC(mx)!=k || AC(mn)!=k) cout<<-1<<endl;
46     else cout<<mn<<" "<<mx<<endl;
```


AC(n)表示给定n时的完成题数

```
16 11 AC(11 n){
17     11 nAC=0;
18     11 nLines=0;
19     for(11 i=1;i<=L;i++){
20         

|  |
|--|
|  |
|  |
|  |
|  |
|  |
|  |
|  |


21
22
23
24
25
26     }
27     return nAC;
28 }
```

太戈编程

2795

杂草种子

请同学写出题目大意
已知什么求什么

01二维表格里,若矩形3个角已有1,剩下角会自动补1,
求人工加几个1可以自动填满表格

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】

10%数据: $n \leq 2, m \leq 2$

50%数据: $n \leq 20, m \leq 20$

所有数据: 保证 $n, m, q \leq 200000$ 。

输入

2 3 2
1 1
2 1

输出?

3

发现+猜想

自动填充完毕后
0格必须花费1次人工填充

人工填充后再自动填充,不断重复

贪心法为什么正确?

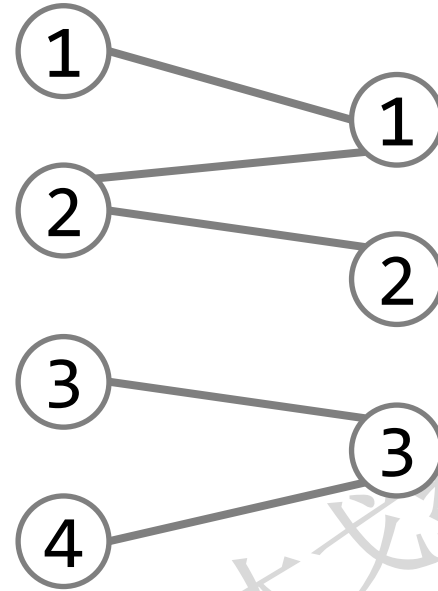
二维01表对应图的连通信息

二分图建模

每行对应左侧节点

每列对应右侧节点

1	0	0
1	1	0
0	0	1
0	0	1



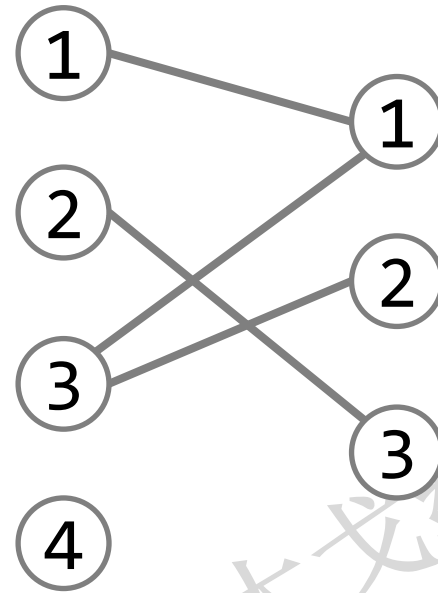
二维01表对应图的连通信息

二分图建模

每行对应左侧节点

每列对应右侧节点

1	0	0
0	0	1
1	1	0
0	0	0



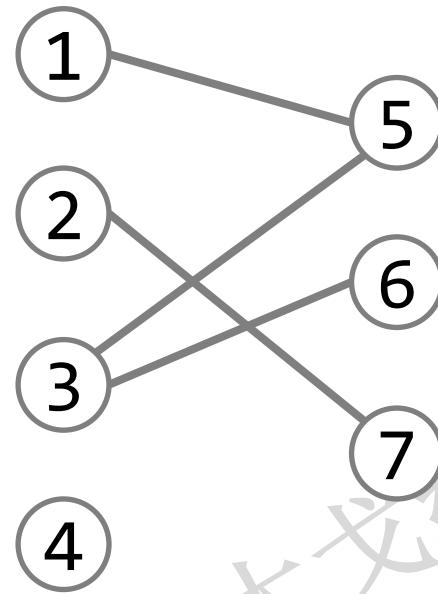
二维01表对应图的连通信息

二分图建模

每行对应左侧节点

每列对应右侧节点

1	0	0
0	0	1
1	1	0
0	0	0



重新
编号

答案是 连通块数-1

```
3  const int N=400009;  
4  bool vst[N];  
5  vector<int> to[N];  
6  void add(int u,int v){  
7      to[u].push_back(v);  
8      to[v].push_back(u);  
9  }
```

```
20 int r,c,q;
21 cin>>r>>c>>q;
22 for(int i=1;i<=q;i++){
23     int x,y;
24     cin>>x>>y;
25     add(x,r+y);
26 }
```

```
27 int nV=r+c;
28 int cnt=0;
29 for(int u=1;u<=nV;u++)
30     if(!vst[u]){
31         dfs(u);
32         cnt++;
33     }
34 cout<<cnt-1<<endl;
```

```
10 void dfs(int u){
11     vst[u]=1;
12     for(int i=0;i<to[u].size();i++){
13         int v=to[u][i];
14         if(!vst[v]) dfs(v);
15     }
16 }
```

太戈编程

2739