

# C++

day2 SH-B260102 集训-讲解

# T1 躲猫猫

请同学写出题目大意  
已知什么求什么

B在数轴某处,有N条信息描述  
每条信息,要么说 $B \leq p_i$   
要么说 $B \geq p_i$   
有些信息是假的  
求假信息的最少数量

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】

$1 \leq N \leq 1000$ 。

$0 \leq p_i \leq 10^9$ 。

莱奥编程

| 输入样例            |
|-----------------|
| 2<br>G 3<br>L 5 |
| 输出样例            |
| 0<br>有可能没有假信息   |

| 输入样例                    |
|-------------------------|
| 2<br>G 3<br>L 2         |
| 输出样例                    |
| 1<br>不能同时为真<br>至少有一条假消息 |

## 数学性质

若  $x \leq L_1$  为真, 且  $L_1 \leq L_2$ , 则  $x \leq L_2$  为真

若  $x \geq G_1$  为真, 且  $G_2 \leq G_1$ , 则  $x \geq G_2$  为真

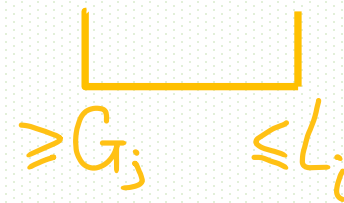
若  $G \leq x \leq L$ , 必须  $G \leq L$ , 否则  $x$  无解

# 算法分析

目标: 兼容的信息条数最大化

将 $\leq$ 关系加入L数组  
将 $\geq$ 关系加入G数组  
对L和G按从小到大排序

哪怕没有一对 $L_i$ 和 $G_j$ 同真  
同时为真的信息条数也至少是  
 $\max(\text{cntL}, \text{cntG})$



枚举所有 $L_i$ 和 $G_j$ 组合  
只要 $G_j \leq L_i$   
说明至少这一对关系可同真

此时, **至少**  $L_i, L_{i+1}, \dots, L_{\text{cntL}}$  同时为真  
且 **至少**  $G_1, G_2, \dots, G_j$  同时为真

此时为真的关系数 **至少** 是  $(\text{cntL} - i + 1) + j$

可能有一些 $G_{p(p>j)}$ 和 $L_{q(q<i)}$ 仍可同时满足  
但当枚举到所有 $L_{i'} (i' \leq i)$ 和 $G_{j'} (j' \geq j)$ 的组合  
总有一个时候  
这两段中间没有其他关系

7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

```
cin>>n;
for(int i=1;i<=n;i++){
    cin>>ch>>t;
    if(ch=='L') L[++cntL]=t;
    else G[++cntG]=t;
}
sort(L+1,L+cntL+1);
sort(G+1,G+cntG+1);
int OK=max(cntL,cntG);
for(int i=1;i<=cntL;i++)
    for(int j=1;j<=cntG;j++)
        if(L[i]>=G[j]) OK=max(OK, );
        else ;
cout<< <<endl;
```

$\geq G_j$      $\leq L_i$

最多真消息数ans  
初始化为序列L和G的  
长度的较大者

菜力编程

# T2 消磨假期

莱奥编程

## 题意

已知 $m$ 册小说的欢乐值，每天看若干册，晚上减半，求 $n$ 天内欢乐值的最小值至少是多少

### 【数据规模与约定】

对于 30%的数据， $1 \leq n \leq 7$ ， $1 \leq m \leq 7$ ；

对于 60%的数据， $1 \leq n \leq 500$ ， $1 \leq m \leq 500$ ；

对于 100%的数据， $1 \leq n \leq 100000$ ， $1 \leq m \leq 100000$ ； $1 \leq a_i \leq 20000$ 。

输入

输出?

4 6  
1 5 3 3 4 6

7

最小值最大化问题  
是经典问题

判断可行性+二分


莱奥编程

是否每天的欢乐值能在hp以上

可行性分析

```
7 bool ok(int hp){
8     ll sum=0;
9     for(int i=1,j=1;i<=n;i++){
10         while(sum<hp){
11             if( ) return 0;
12             sum+=a[j];
13             j++;
14         }
15         sum/=2;
16     }
17     return 1;
18 }
```

莱奥编程

```
22     cin>>n>>m;
23     ll A=0;
24     for(int i=1;i<=m;i++) {
25         cin>>a[i];
26         A+=a[i];
27     }
28     ll l=1;
29     ll r=A;
30     ll ans=1;
31     while(l<=r){
32         
33
34
35     }
36     cout<<ans<<endl;
37     return 0;
38 }
```

# T3 单词构造

莱奥编程

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 }
```

葉奧编程

## 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 }
```

莱奥编程

```
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

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

菜奥编程

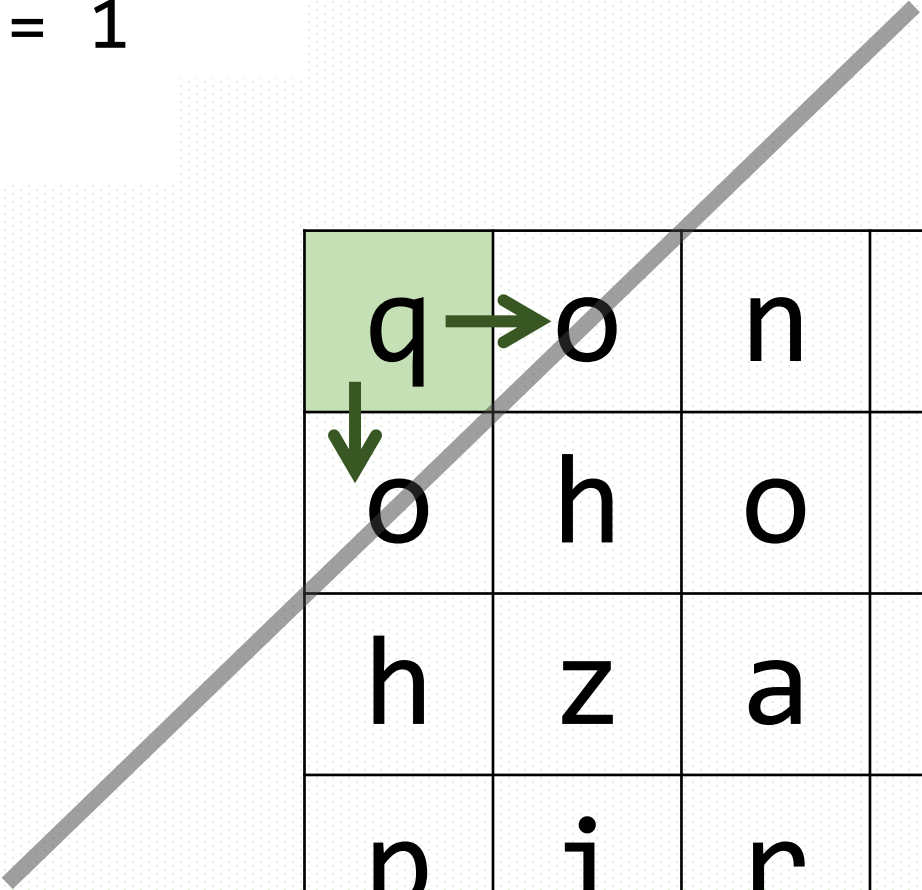
step = 1

(1, 1)

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

贪心  
变种



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

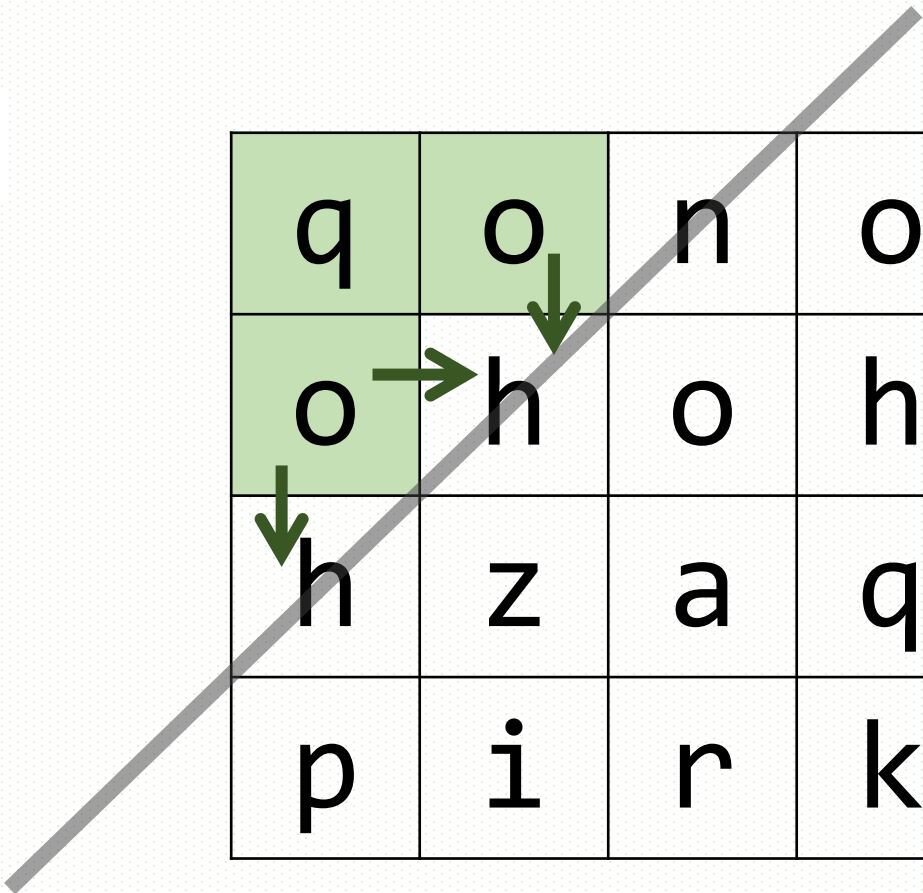
答案字符串 = q???????

step = 1

(1,1)

step = 2

(1,2) (2,1)



|   |   |   |   |   |
|---|---|---|---|---|
| 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 = qo??????

step = 1

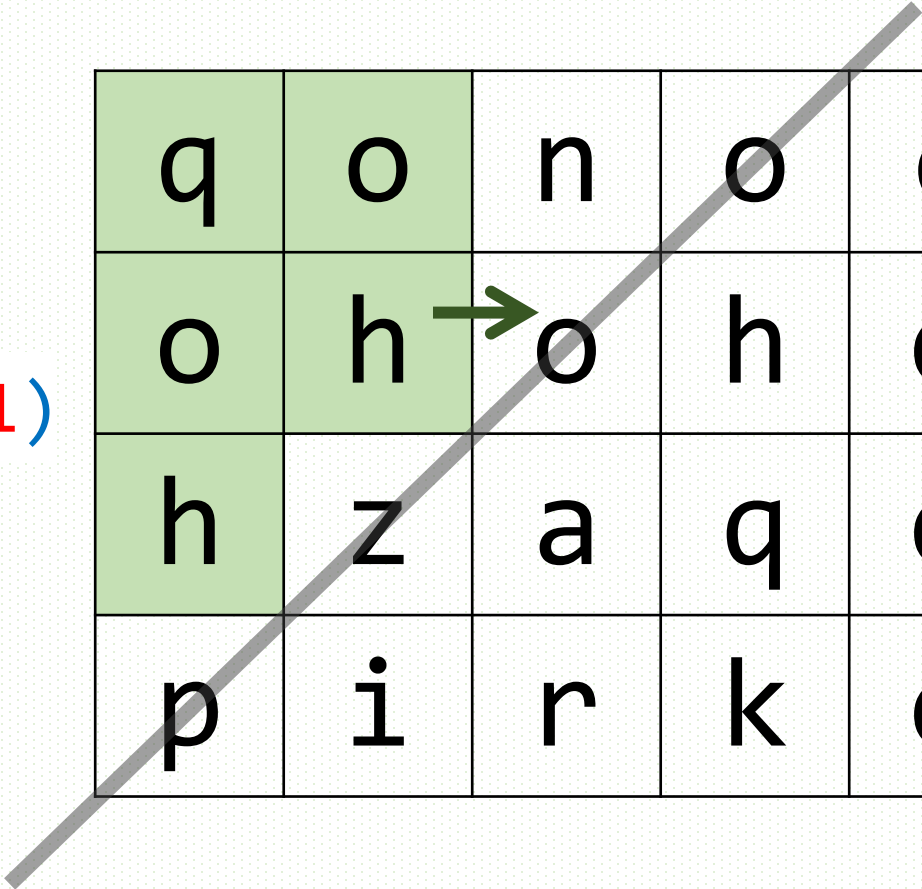
(1,1)

step = 2

(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)



|   |   |   |   |   |
|---|---|---|---|---|
| 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 = qoh?????

step = 1

(1,1)

step = 2

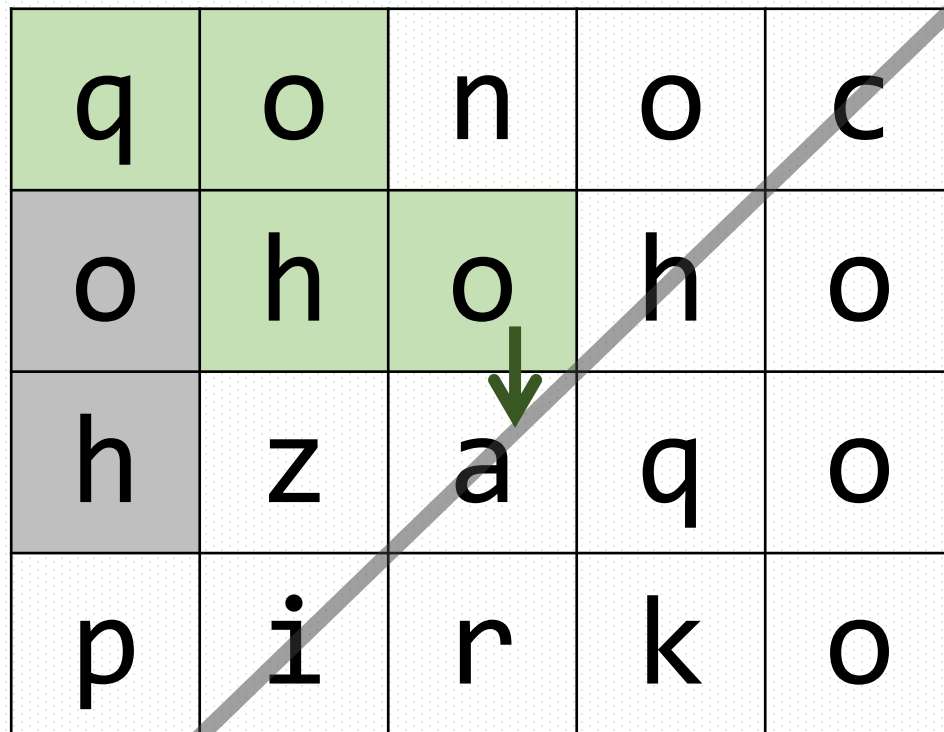
(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)

step = 4

(2,3) (3,2) (4,1)



|   |   |   |   |   |
|---|---|---|---|---|
| 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 = qoho????

step = 1

(1,1)

step = 2

(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)

step = 4

(2,3) (3,2) (4,1)

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

step = 5

(2,4) (3,3)

答案字符串 = qohoa???

step = 1

(1,1)

step = 2

(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)

step = 4

(2,3) (3,2) (4,1)

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

step = 5

(2,4) (3,3)

step = 6

(3,4) (4,3)

答案字符串 = qohoaq??

step = 1

(1,1)

step = 2

(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)

step = 4

(2,3) (3,2) (4,1)

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

step = 5

(2,4) (3,3)

step = 6

(3,4) (4,3)

step = 7

(3,5) (4,4)

答案字符串 = qohoaqk?

step = 1

(1,1)

step = 2

(1,2) (2,1)

step = 3

(1,3) (2,2) (3,1)

step = 4

(2,3) (3,2) (4,1)

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

step = 5

(2,4) (3,3)

step = 6

(3,4) (4,3)

step = 7

(3,5) (4,4)

step = 8

(4,5)

答案字符串 = qohoaqko

贪心  
变种

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

枚举步长

枚举行

计算对应y坐标

取最小的字符

当前位置是否是选择的位置?

从当前位置向下、向右，  
标记已访问

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 }
```

# T4 风水大师

莱奥编程

给定 $n$ 个物品的 $m$ 条对它们属性的相生相克判断，求矛盾的判断数量

如何记录相生相克关系？

带权并查集

【数据规模与约定】

| 测试点编号 | 特殊性质                            |
|-------|---------------------------------|
| 1     | 全为克制关系判断, $1 \leq n, m \leq 10$ |
| 2     | 全为相生关系判断, $1 \leq n, m \leq 10$ |

考虑特殊数据

全为克制

全为相生

莱奥编程

# 并查集

查找+合并

建立最方便查找的双向关系

是否等价/相连/有关

=

是否有相同的根

相生相克关系

不是双向关系，单向且有层级

如何快速判断两个物品之间的

单向关系：

大于/小于/吃/被吃/生/克

# 带权并查集

查找+合并

建立最方便查找的单向关系

是否大于/小于/  
吃/被吃/生/克

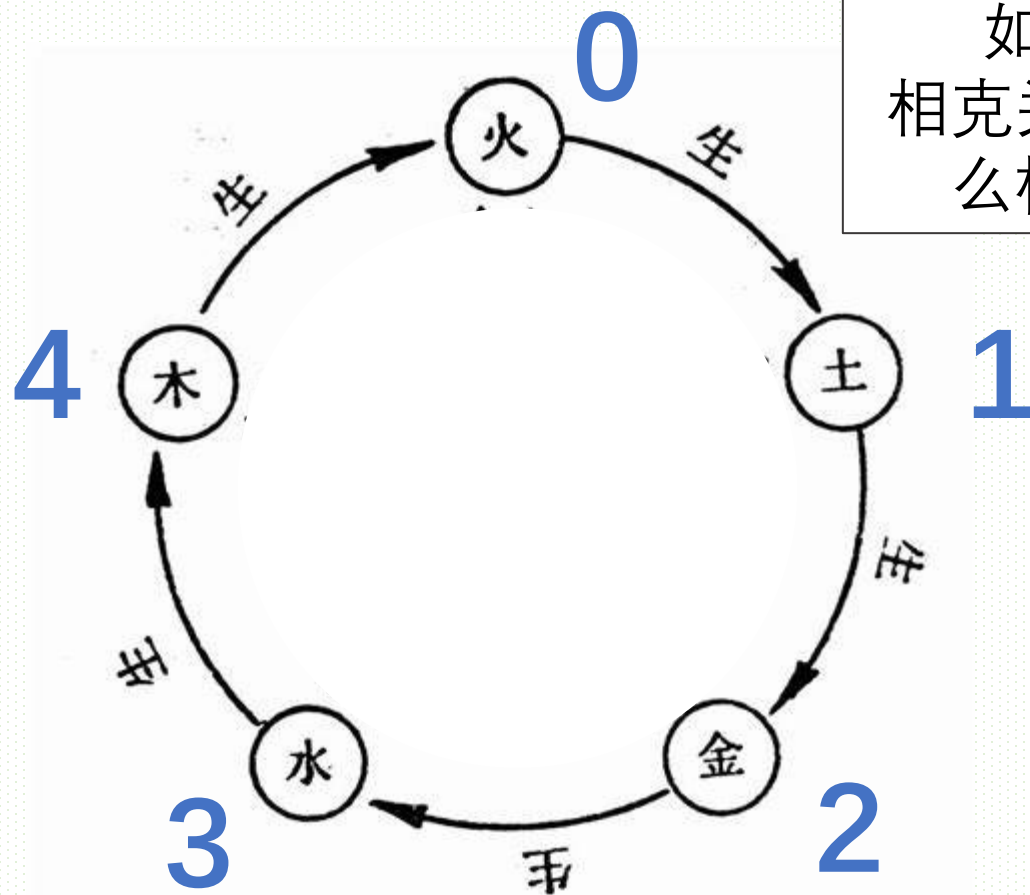
比较节点和根之间的  
距离（权值）

带权并查集使用两个数组记录关系

$p[i]$ 数组记录  
父亲关系

$w[i]$ 数组记录  
和根的距离=权值

# 相生关系

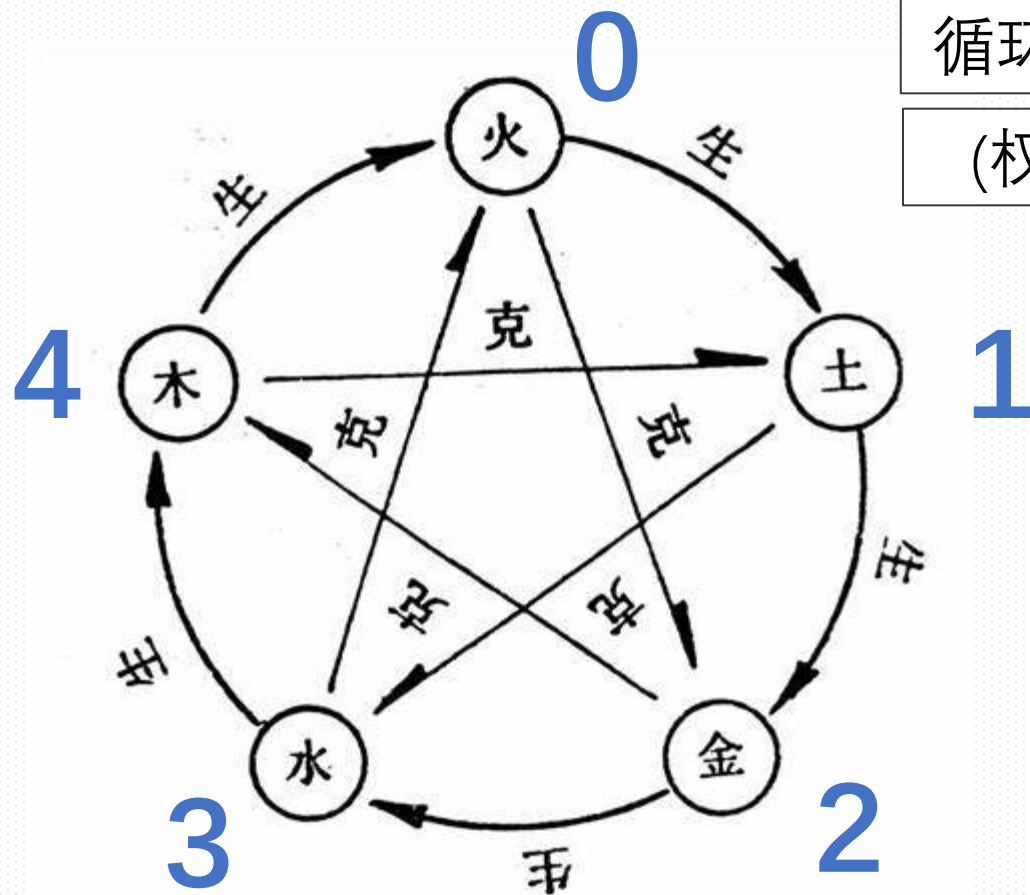


如果只有  
相克关系图是什  
么样的呢?

5个属性地位相同，任意属性都能做根

沿“生”方向给每个属性赋权

# 相生相克关系



循环了怎么办

$(\text{权值} + d) \% 5$

观察相生相克与权值的关系

生=+1, 克=+2

# 带权并查集

权值

5个属性依次编号0, 1, 2, 3, 4

“查”

新一条判断中两个物品是否已有关系=是否共享一个根

已有关系：和新判断是否矛盾：权值是否符合要求

没有关系：建立关系并更新权值

“并”

建立新判断中的两个物品的根之间的联系并更新权值

莱奥编程

# 矛盾判断逻辑

$$a \text{到} b \text{的权值} = (w[a] - w[b]) \% 5$$

$$b \text{到} a \text{的权值} = (w[b] - w[a]) \% 5 = -(w[a] - w[b]) \% 5$$

输入的关系 **d** 表示 **b到a** 的权值

$$\text{所以应该满足: } -(w[a] - w[b]) \% 5 == d \% 5$$

$$\text{等价于: } (w[a] - w[b]) \% 5 == -d \% 5$$

$$\text{化简为: } (w[a] + d) \% 5 == w[b]$$

$$\text{所以矛盾条件为: } (w[a] + d) \% 5 \neq w[b]$$



```
int p[N],w[N];
```

p[i]: 并查集父节点数组, i所在集合的根节点点

w[i]: i到其集合根节点的权值

```
int d=(ch=='k'?2:1);
```

新的相生相克关系判断, b到a的权值

```
int ra=root(a),rb=root(b);
```

```
if(ra==rb)
```

如果a和b已经有关系(在一个集合中)

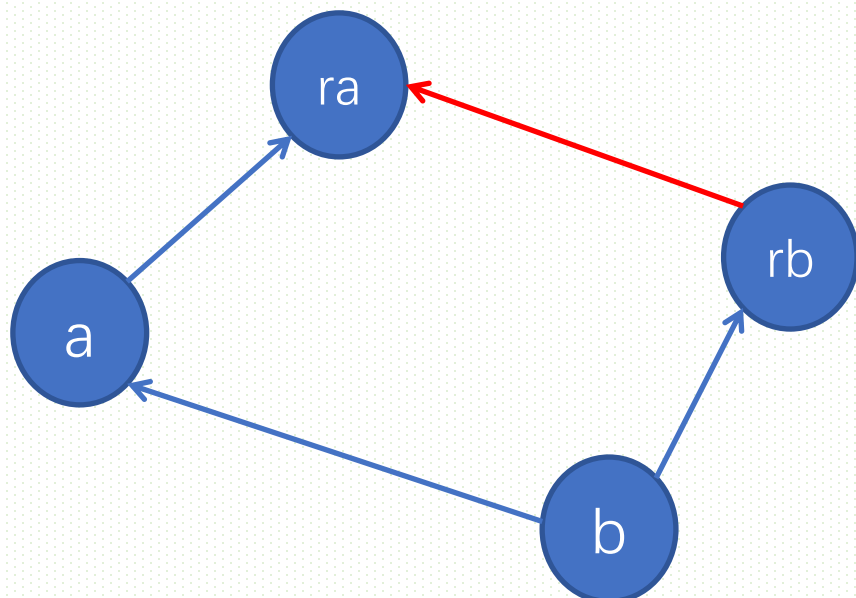
```
ans+=((w[a]+d)%5!=w[b]);
```

判断与已有关系是否矛盾

```
else{
    p[rb]=ra;
}

```

“并”=建立根节点之间的联系，  
合并后，ra所在集合的根节点是rb.  
需要更新rb的权值 **如何更新**



合并后，需要满足：a到b的权值 = -d

而a到b的权值可以通过路径a->ra->rb->b计算

a到ra:  $w[a]$

ra到rb:  $-w[rb]$  ( $w[rb]$ 是rb到ra的权值)

rb到b:  $-w[b]$  ( $w[b]$ 是b到rb的权值)

所以:  $w[a] - w[rb] - w[b] = -d \pmod{5}$

—>:  $w[rb] = w[a] - w[b] + d \pmod{5}$

$w[rb] = (w[a] + d - w[b] + 5) \% 5$

**另一种解释**: 建立rb指向ra的关系之后，  
b到ra的两条路径应该长度相等

$d + w[a] = w[b] + w[rb]$

# T5 股神七

莱奥编程

# 股神七

你是股神，你的神力是能看到未来 $n$ 天里每天股票的盈利或者亏损，第 $i$ 天盈利 $x[i]$ 元，当然如果 $x[i]$ 是负数，代表亏损。你可以在一段连续的日期进行投资，为了盈利不少于 $m$ 元，至少需要投资连续的几天？若无法盈利 $m$ 元，输出0

输入样例

5 10000

1 1 1 1 1

输出样例

0

输入样例

3 1

-100 2 99

输出样例

1

输入样例

5 6

2 2 2 3 3

输出样例

2

暴力该怎么做？

自编哪几套测试  
数据才够全面？

## TE算法1：枚举左右端点+前缀和

```
12  int ans=n+1;
13  for(int i=1;i<=n;i++)
14      for(int j=i;j<=n;j++)
15          if(sum[j]-sum[i-1]>=m){
16              ans=min(ans,j-i+1);
17              break;
18          }
19  cout<<(ans==n+1?0:ans)<<endl;
```

## TE算法1： 枚举左右端点+前缀和

```
12  int ans=n+1;  
13      枚举左端点位置i  
14      枚举右端点位置j  
15      if(sum[j]-sum[i-1]>=m){  
16          ans=min(ans,j-i+1);  
17          break;  
18      }  
19  cout<<(ans==n+1?0:ans)<<endl;
```

## TE算法2：枚举答案+右端点+前缀和

```
12  int ans=0, END=0;
13  for(int len=1; len<=n && !END; len++)
14      for(int i=len; i<=n; i++)
15          if(sum[i]-sum[i-len]>=m){
16              ans=len; END=1;
17              break;
18          }
19  cout<<ans<<endl;
```

# AC算法3： 二分枚举答案 $O(N\log N)$

二分枚举投资天数 $k$

判断连续 $k$ 天内能否赚 $m$ 元

子问题

最多 $k$ 天连续投资  
求最大盈利

连续子段最大和： 长度不超过 $k$

# 连续子段最大和： 长度不超过k

$f[i]$ 代表以*i*号数结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = \max \left\{ \begin{array}{l} x[i], \\ x[i] + x[i-1], \\ x[i] + x[i-1] + x[i-2], \\ \dots \dots \dots \\ x[i] + x[i-1] + \dots + x[i-k+1] \end{array} \right\}$$

长度为1

长度为2

长度为3

长度为k

注意越界

# 连续子段最大和： 长度不超过k

$f[i]$ 代表以i号数结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = \max \left\{ \begin{array}{l} s[i] - s[i-1], \\ s[i] - s[i-2], \\ s[i] - s[i-3], \\ \dots \dots \dots \\ s[i] - s[i-k] \end{array} \right\}$$

连续和  
转换为  
前缀和

长度为1

长度为2

长度为3

长度为k

注意越界

# 连续子段最大和： 长度不超过k

$f[i]$ 代表以*i*号数结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = s[i] - \min \left\{ \begin{array}{l} s[i-1], \\ s[i-2], \\ s[i-3], \\ \dots\dots\dots \\ s[i-k] \end{array} \right\}$$

固定*i*时  
前缀和数组  
窗口最小值

单调队列

注意越界

$s[i]$ 表示x数组1号到i号的总和

$MIN[i]$ 表示以 $s[i]$ 为结尾  
长度不超过k的窗口最小值

$$f[i] = s[i] - MIN[i-1]$$

$$f[i] = s[i] - \min \left\{ \begin{array}{l} s[i-1], \\ s[i-2], \\ s[i-3], \\ \dots \dots \dots \\ s[i-k] \end{array} \right\}$$

注意越界

莱奥编程

```

6 bool OK(int k){
7     int l=1,r=1,MAX=0;
8     for(int i=1;i<=n;i++){
9         while(l<r&&sum[i]-sum[l]>=k+1)l++;
10        while(l<r&&sum[r]-sum[l]<sum[i]-sum[l])r++;
11        q[r++]=i;
12        MAX=max(MAX,sum[i]-sum[q[l]]);
13        if(MAX>=m)return 1;
14    }
15    return 0;
16 }

```

i=0

为什么错

```

23 int l=0,r=n,ans=n+1;
24 while(l<=r){
25     int mid=l+(r-l)/2;
26     if(OK(mid))ans=mid,r=mid-1;
27     else l=mid+1;
28 }
29 cout<<(ans==n+1?0:ans)<<endl;

```

讨论：能否做到  
O(n)复杂度？

拓展思考。

# SH-B260102 订正题号

T1 3186 躲猫猫

T2 2808 消磨假期

T3 2720 单词构造

T4 3484 风水大师

T5 652 股神七

莱奥编程