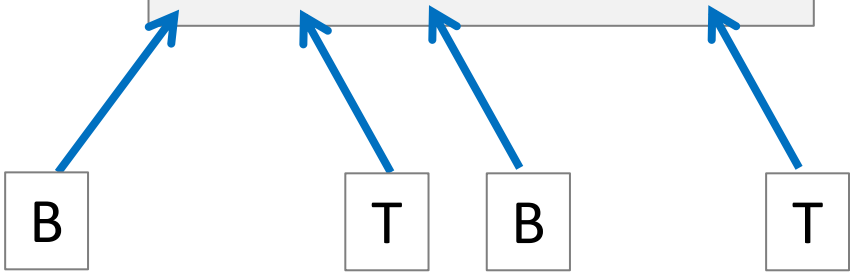
信奥算法

1790

输入

```
10 16
TTTTTTTTT**TBBBT
BTTTTTTTTTBBBBBB
BBBBTTTBTTBBBBBB
BBBBBTTTTTTTTTTB
BBBBBTTBBBBTTTBB
BBBBBTTTTTTTTBBT
TBBBBTTTBBTTTTTB
TTBTBBBBBBBTTBTB
TTTTBBBBBTBTTTTT
BTTTTBBBTTBBBTTT
```

输出几?



贪心法

题目要求人数最少

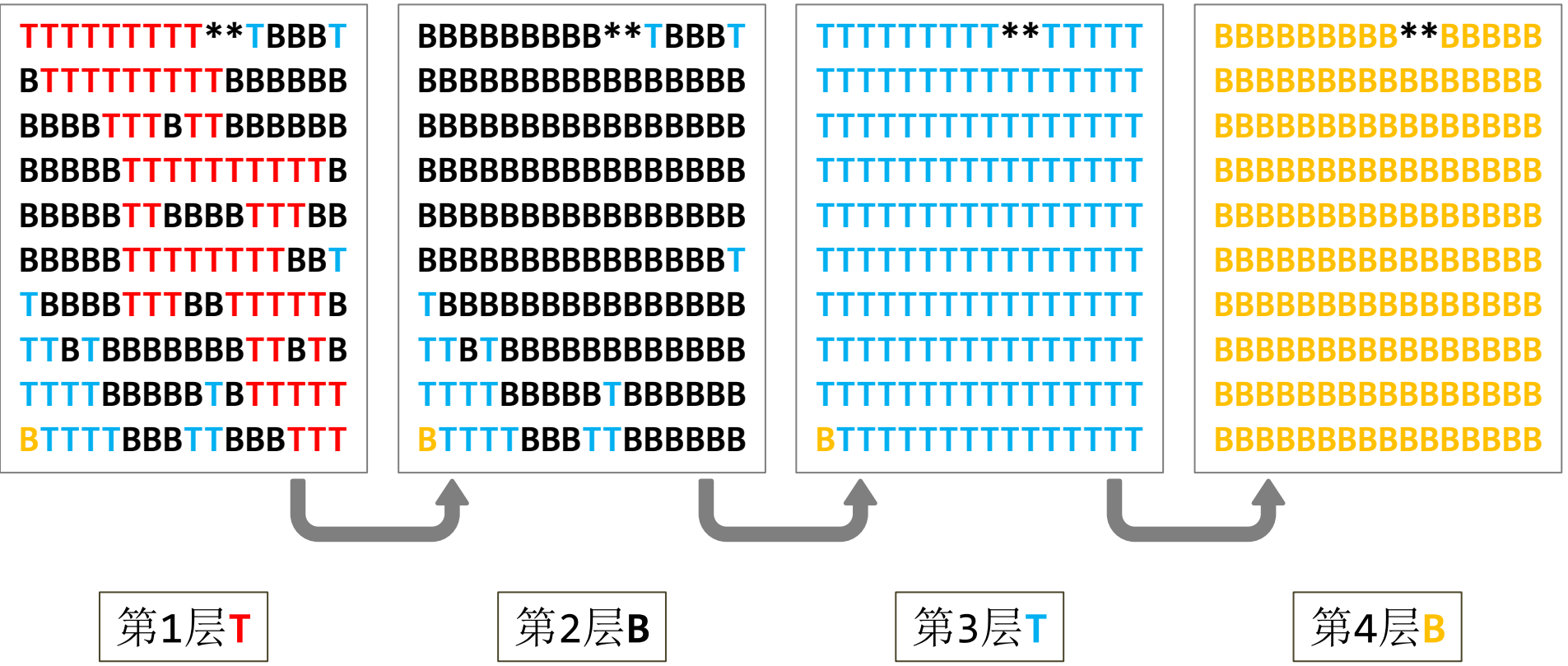
决策问题	每格脚印是第几人踩出来的 脚印层次的分配决策
贪心求解	从边边角角开始决策
	当前左上角和右下角一定是最上层
	每次找可能是最上层的所有脚印 全部当做最上层
决策问题	被最上层覆盖掉的是什么信息时 对应的人数最少?
贪心求解	最上层直接覆盖的第二层格子 均为不同于最上层的另一种脚印

贪心法
实现1

75分

迭代+洪水填充

单次洪水	将当前左上角到右下角的脚印换成另一种	迭代多次	终止条件是什么?



第1层T　　第2层B　　第3层T　　第4层B

贪心法
实现1

迭代+洪水填充

75分

```
22  cin>>R>>C;
23  for(int i=1;i<=R;++i)
24      for(int j=1;j<=C;++j){
25          cin>>d[i][j];
26          if(d[i][j]!='*')cnt++;
27      }
28  int ans=0;
29  do{
30      ++ans;
31      area=0;
32      dfs(1,1);
33  }while
34  cout<<ans<<endl;
```

cnt记录共有几格脚印

area记录此次洪水填充到几格

迭代条件

贪心法
实现1

洪水填充

75分

```
 8 void dfs(int x,int y){
 9     area++;
10     char pre=d[x][y];
11     d[x][y]=pre=='T'?'B':'T';
12     for(int k=0;k<4;k++){
13         int nx=x+dx[k],ny=y+dy[k];
14         char nxt=d[nx][ny];
15         if(nx>=1&&nx<=R&&ny>=1&&ny<=C
16             dfs(nx,ny);
17     }
18 }
```

DFS复杂度	最差情况迭代次数	程序总体复杂度
O(RC)	O(RC)	$O(R^2C^2)$

贪心法
实现2

建图+BFS求层数

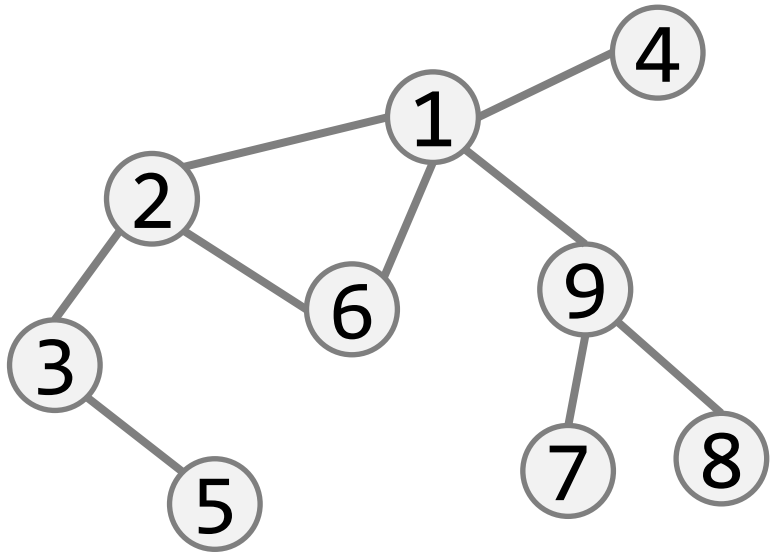
100分

图论
建模

每个连通块缩成1个节点	重新编号
若2个连通块相邻就将对应节点连边	

```
TTTTTTTT**TBBBT
BTTTTTTTTTBBBBBB
BBBBTTTBTTBBBBBB
BBBBBTTTTTTTTTTB
BBBBBTTTTTTTTTBB
BBBBBTTTTTTTTBBB
TBBBBTTTBBTTTTTT
TTBTBBBBBBBTTTTT
TTTTBBBBBTTTTTTT
BTTTTBBBTTBBBTTT
```

```
111111111**79998
211111111199999
222211141199999
222221111111119
222221111111199
222221111111999
322221112211111
332322222211111
333322222661111
533332226622211 1
```



程序总体复杂度?
O(RC)

贪心法
实现2

建图+BFS求层数

100分

id[x][y]记录(x,y)格子 所在连通块对应的块号
id[x][y]为0代表 (x,y)格子还没有访问过

贪心法
实现2

建图+BFS求层数

100分

```
65      for(int i=1;i<=R;++i)
66          for(int j=1;j<=C;++j)
67              cin>>d[i][j];
68      for(int i=1;i<=R;++i)
69          for(int j=1;j<=C;++j)
70              if(d[i][j]!='*')
71                  cnt++;
72                  dfs(i,j);
73              }
74      build();
75      bfs();
```

每个连通块缩成1个节点	
重新编号	cnt记录当前连通块号
号码1,2,3,...	

贪心法
实现2

建图+BFS求层数

100分

```
52 void dfs(int x,int y){
53
54     for(int k=0;k<4;++k){
55         int nx=x+dx[k],ny=y+dy[k];
56         if(nx<1||nx>R||ny<1||ny>C)continue;
57         if(d[nx][ny]==d[x][y]&&!id[nx][ny])
58             dfs(nx,ny);
59     }
60 }
```

id[x][y]记录(x,y)格子 所在连通块对应的块号
id[x][y]为0代表 (x,y)格子还没有访问过

贪心法
实现2

建图+BFS求层数

100分

```
 8 vector<vector<int> > to;
 9 void build(){
10     to.resize(cnt+1);
11     for(int x=1;x<=R;++x)
12         for(int y=1;y<=C;++y){
13             if(d[x][y]=='*')continue;
14             for(int k=0;k<4;++k){
15                 int nx=x+dx[k],ny=y+dy[k];
16                 if(nx<1||nx>R||ny<1||ny>C)continue;
17                 if(d[nx][ny]=='*')continue;
18                 if(d[nx][ny]==d[x][y])continue;
19
20             }
21         }
22 }
```

动态扩充

贪心法
实现2

建图+BFS求层数

100分

```
23 vector<int> level,vst;
24 void bfs(){
25     queue<int> q;
26     level.resize(cnt+1); vst.resize(cnt+1);
27     q.push(1);  vst[1]=1;  level[1]=1;
28     int ans=1;
29     while(!q.empty()){
30         int u=q.front(); q.pop();
31         for(int i=0;i<to[u].size();++i){
32             int v=to[u][i];
33             if(vst[v])continue;
34             vst[v]=1;
35             ans=level[v]=level[u]+1;
36             q.push(v);
37         }
38     }
39     cout<<ans<<endl;
40 }
```

1856

请同学写出题目大意
已知什么求什么

n*m棋盘格，每格是小写字母a到z其中之一。如两格四向相邻且字母相同则属于同一部落。q种矩形，问每个矩形内涉及到几个部落？

有提示作用	对于10%数据，保证n=1。
有提示作用	对于10%数据，保证只有1种宝石。
	对于50%数据，保证n,m,q<=500。
	对于100%数据，保证n,m<=2000,q<=5000。

10分

只有1种宝石

```
#include<bits/stdc++.h>
using namespace std;
const int N=2009;
int n,m,q;
char d[N][N];
int main() {
    freopen("king.in","r",stdin);
    freopen("king.out","w",stdout);
    cin>>n>>m;
    for(int i=1;i<=n;++i)
        for(int j=1;j<=m;++j)
            cin>>d[i][j];
    cin>>q;
    while(q--)cout<<1<<endl;
    return 0;
}
```

d[i][j]表示(i,j)格的字母

40分

模拟+暴力

按照题意形成直观的算法步骤

标记部落	如两格四向相邻 且字母相同则属于同一部落
DFS/BFS	多次洪水填充标记每块部落 当前左上角和右下角一定是最上层

```
25      for(int i=1;i<=n;++i)
26          for(int j=1;j<=m;++j)
27              if(!id[i][j]){
28                  ++nID;
29                  dfs(i,j);
30              }
```

id[i][j]表示(i,j)格属于几号部落

nID表示当前部落数量

40分

模拟+暴力

```
 8  void dfs(int x,int y){
 9      id[x][y]=nID;
10      for(int k=0;k<4;++k){
11          int nx=x+dx[k],ny=y+dy[k];
12          if(nx<1||nx>n||ny<1||ny>m)continue;
13          if(id[nx][ny])continue;
14          if(                        )continue;
15          dfs(nx,ny);
16      }
17  }
```

40分

模拟+暴力

```
36      for(int i=1;i<=q;++i){
37          int r1,c1,r2,c2;
38          cin>>r1>>c1>>r2>>c2;
39          set<int> s;
40          for(int x=r1;x<=r2;++x)
41              for(int y=c1;y<=c2;++y)
42                  s.insert(id[x][y]);
43          cout<<s.size()<<endl;
44      }
```

如何想到加速算法

用简化问题启发灵感

二维棋盘格如何简化?

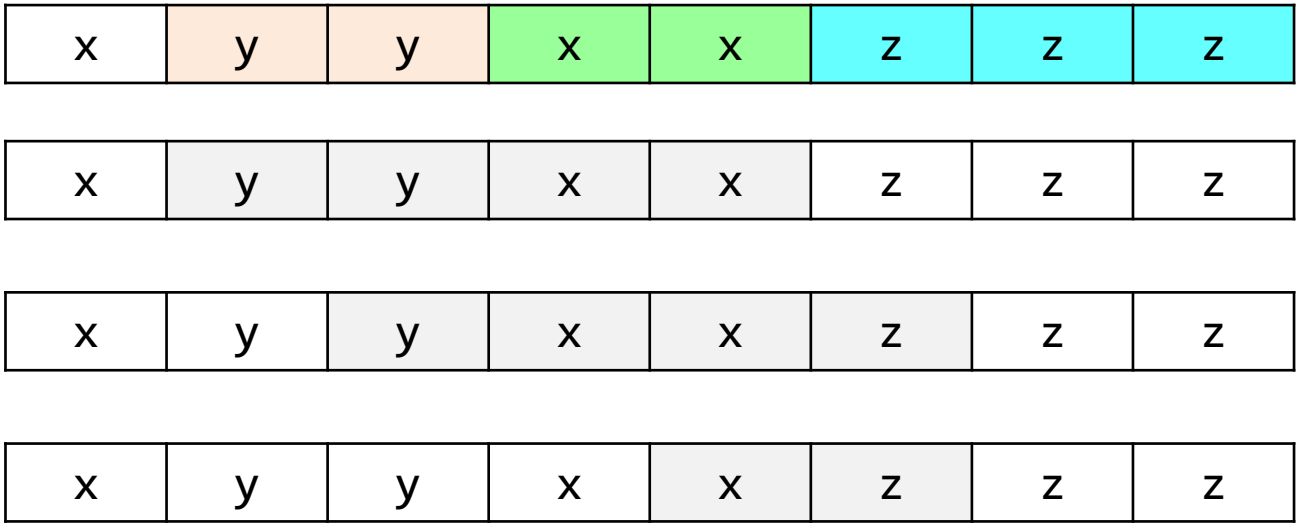
对于10%数据，保证n=1。

10分

一维问题

n个字母序列，如两格相邻且字母相同则属于同一部落。q种区间，问每个区间内涉及到几个部落?

想出尽量快的算法



选代表计数法

部落连续段记作1份

选取1个代表

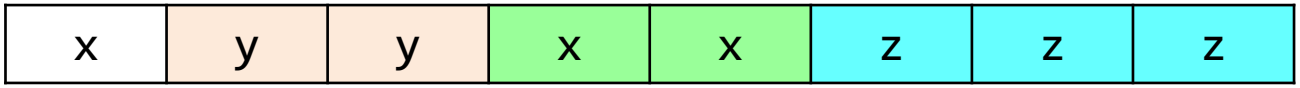
例如部落左端点

答案约等于区间内有几个部落代表

10分

一维问题

n个字母序列，如两格相邻且字母相同则属于同一部落。q种区间，问每个区间内涉及到几个部落?



选代表计数法

部落连续段记作1份

选取1个代表

例如部落左端点

答案约等于区间内有几个部落代表

10分

一维问题

```
35 int main() {
36     freopen("king.in","r",stdin);
37     freopen("king.out","w",stdout);
38     cin>>n>>m;
39     if(n==1) solve();
40     else ;
41     return 0;
42 }
```

10分

一维问题

```
6   int rep[N],id[N];
7   bool ok[N];
8   char d[N];
9   for(int j=1;j<=m;++j)cin>>d[j];
10  int nID=1;
11  rep[1]=id[1]=ok[1]=1;
12  for(int j=2;j<=m;++j)
13      if(d[j]==d[j-1]) id[j]=id[j-1],ok[j]=0;
14      else  id[j]=++nID,rep[nID]=j,ok[j]=1;
```

id[j]表示j号格属于几号部落

nID表示当前部落数量

rep[x]表示x号部落的代表是几号格

ok[j]表示j号格是否为某部落的代表

10分

一维问题

```
15    int s[N];
16    s[0]=0;
17    for(int j=1;j<=m;++j)s[j]=s[j-1]+ok[j];
```

ok[j]表示j号格是否为某部落的代表

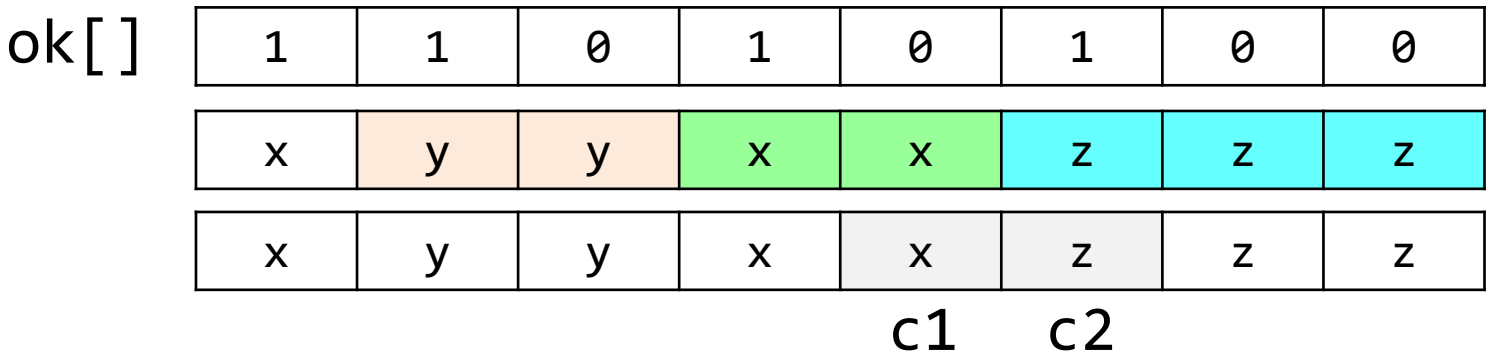
ok[]是01数组
s[]是ok[]的前缀和

s[j]表示1到j号格共有几个部落代表

10分

一维问题

```
18      cin>>q;
19      for(int i=1;i<=q;++i){
20          int r1,c1,r2,c2;
21          cin>>r1>>c1>>r2>>c2;
22          int ans=s[c2]-s[c1-1];
23          if(rep[id[c1]]<c1)ans++;
24          cout<<ans<<endl;
25      }
```



如何想到加速算法

请总结一维算法

思考如何推广到二维情况

100分

代表计数+前缀和+边框处理

id[x][y]表示(x,y)格属于几号部落

nID表示当前部落数量

```
struct Node{int x,y;};
Node rep[N*N];
```

rep[i]表示i号部落的代表是哪个格子

ok[x][y]表示(x,y)格是否为某部落的代表

s[x][y]表示(1,1)到(x,y)格共有几个部落代表

100分

代表计数+前缀和+边框处理

问询：某矩形内共几个部落
答案分为2部分

1

矩形内包含几个部落代表
二维连续和RSQ用二维前缀和计算

2

矩形内有几个部落的代表不在矩形内
这些部落一定在矩形边框上
查看矩形边框上所有格子 找该格所属部落的代表 判断是否在矩形内

100分

代表计数+前缀和+边框处理

```
41      for(int i=1;i<=n;++i)
42          for(int j=1;j<=m;++j)
43              if(!id[i][j]){
44                  ++nID;
45                  rep[nID]=(Node){i,j};
46                  ok[i][j]=1;
47                  dfs(i,j);
48              }
```

id[x][y]表示(x,y)格属于几号部落
nID表示当前部落数量
rep[i]表示i号部落的代表是哪个格子
ok[x][y]表示(x,y)格是否为某部落的代表

100分

代表计数+前缀和+边框处理

```
49      for(int i=1;i<=n;++i)
50          for(int j=1;j<=m;++j)
51              s[i][j]=s[i][j-1]+s[i-1][j]
52                      -s[i-1][j-1]+ok[i][j];
```

ok[x][y]表示(x,y)格是否为某部落的代表

s[x][y]表示(1,1)到(x,y)格共有几个部落代表

100分

代表计数+前缀和+边框处理

```
int r1,c1,r2,c2;
cin>>r1>>c1>>r2>>c2;
int nRepIn=s[r2][c2]+s[r1-1][c1-1]
          -s[r1-1][c2]-s[r2][c1-1];
```

矩形内包含几个部落代表

100分

代表计数+前缀和+边框处理

```
31 bool out(Node&o,int&r1,int&c1,int&r2,int&c2){
32     return o.x<r1||o.x>r2||o.y<c1||o.y>c2;
33 }
```

判断格子o是否在矩形外部

100分

代表计数+前缀和+边框处理

2

矩形内有几个部落的代表不在矩形内
这些部落一定在矩形边框上
查看矩形边框上所有格子 找该格所属部落的代表 判断是否在矩形内

100分

```
set<int> RepOut;
int y=c1;
for(int x=r1;x<=r2;++x)
    if(out(rep[id[x][y]],r1,c1,r2,c2))
        RepOut.insert(id[x][y]);
y=c2;
for(int x=r1;x<=r2;++x)
    if(out(rep[id[x][y]],r1,c1,r2,c2))
        RepOut.insert(id[x][y]);
int x=r1;
for(int y=c1;y<=c2;++y)
    if(out(rep[id[x][y]],r1,c1,r2,c2))
        RepOut.insert(id[x][y]);
x=r2;
for(int y=c1;y<=c2;++y)
    if(out(rep[id[x][y]],r1,c1,r2,c2))
        RepOut.insert(id[x][y]);
int ans=
```

太戈编程

1790,1856