解题报告

A. 双剑合并

题意

从两组数中分别找出一个数，使得这两个数的异或值最大。
n<=1e^5,a[i]<=1e^9

分析

这个数据范围暴力肯定是不行的了，我们肯定需要枚举一列数，如何处理另一列数呢？可以用字典树把按位另一列数存起来，这样就可以做到很快的查询了。
wa了好多次，因为按位取的时候应该从高位到低位，一开始写成了从低位到高位。

思考

思路对了就好，主要还是按位去的时候有个贪心的思想

```
//  Created by Chenhongwei on 2016-05-03 Tuesday 21:51
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
```

```
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e6+100;
typedef long long ll;
typedef unsigned long long ull;
int n,m,pos;
int nxt[maxn*4][2];
int num[maxn*4];
void insert(ll x)
{
        int cur=0;
        for(int i=0;i<=31;i++)
        {
                int flag=(x&(1<<(31-i)))?1:0;
                if(nxt[cur][flag]==-1)
                {
                        memset(nxt[pos],-1,sizeof nxt[pos]);
                        nxt[cur][flag]=pos++;
                }
                cur=nxt[cur][flag];
        }
        num[cur]=x;
}
int find(int x)
{
        int cur=0;
        for(int i=0;i<=31;i++)
        {
                int flag=(x&(1<<(31-i)))?1:0;
                if(nxt[cur][1-flag]!=-1)
                        cur=nxt[cur][1-flag];
                else
                        cur=nxt[cur][flag];
        }
        return x^num[cur];
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
```

```
        int T;
        scanf("%d",&T);
        while(T--)
        {
                memset(num,0,sizeof num);
                scanf("%d%d",&n,&m);
                int maxv=0,tmp;
                pos=1;
                memset(nxt[0],-1,sizeof nxt[0]);
                for(int i=1;i<=n;i++)
                {
                        scanf("%d",&tmp);
                        insert(tmp);
                }
                for(int i=1;i<=m;i++)
                {
                        scanf("%d",&tmp);
                        maxv=max(maxv,find(tmp));
                }
                printf("%d\n",maxv);
        }
        return 0;
}
```

B. 单词替换

题意

一个仅包含小写字母的字符串s，和单词A，B。把s中所有的出现过的A替换为B。

分析

就是简单的字符串匹配，输出的时候替换一下就好了。
中间写的时候，有个地方手残，wa了几发。

思考

主要还是输出的时候注意下就好了

```
//  Created by Chenhongwei on 2016-05-03 Tuesday 23:15
```

```
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=5e6+100;
typedef long long ll;
typedef unsigned long long ull;
char s[maxn],t[maxn],b[maxn];
int nxt[maxn];
void pre_kmp()
{
        memset(nxt,0,sizeof nxt);
        int j=0,m=strlen(t+1);
        for(int i=2;i<=m;i++)
        {
                while(j>0&&t[j+1]!=t[i])
                        j=nxt[j];
                if(t[j+1]==t[i]) j++;
                nxt[i]=j;
        }
}
void kmp()
{
        int j=0,n=strlen(s+1),m=strlen(t+1);
        int last=1;
```

```
        for(int i=1;i<=n;i++)
        {
                while(j>0&&t[j+1]!=s[i])
                        j=nxt[j];
                if(t[j+1]==s[i]) j++;
                if(j==m)
                {
                        // cout<<"ok"<<endl;
                        j=0;
                        for(int k=last;k<=i-m;k++)
                                printf("%c",s[k]);
                        printf("%s",b+1);
                        last=i+1;
                }
        }
        for(int i=last;i<=n;i++)
                printf("%c",s[i]);
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%s%s%s",s+1,t+1,b+1);
                pre_kmp();
                kmp();
                printf("\n");
        }
        return 0;
}
```

C. 01的时间

题意

对于一个数字n，找出它的一个最小倍数，这个倍数只能有0和1构成
T<=500,n<=466

分析

一看到这个数据范围，想到的就是打表，感觉按二进制枚举到ULL就应该够了。这个不是正确思路，一开始的时候没想到这个范围会不够，wa了几次，后来才检查了一下，发现n=396的时候结果出不来，后来特判一下自己找出来结果特判一下

思考

正确思路应该是去搜索，应该和G题差不多

```
//  Created by Chenhongwei on 2016-05-05 Thursday 16:05
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e6+100;
typedef long long ll;
typedef unsigned long long ull;
ull b[maxn];
ull ans[500];
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
```

```
        // freopen("out1.txt","w",stdout);
        for(int i=1;i<(1<<19);i++)
        {
                ll tmp=i&(1<<19);
                for(int j=18;j>=0;j--)
                {
                        int x;
                        x=(i&(1<<j))?1:0;
                        tmp=tmp*10+x;
                }
                b[i]=tmp;
        }
        for(int i=1;i<=466;i++)
        {
                for(int j=1;j<(1<<19);j++)
                        if(b[j]%i==0)
                        {
                                ans[i]=b[j];
                                // printf("%lld\n",ans[i]);
                                break;
                        }
        }
        int T,n;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%d",&n);
                if(n==396)
                        cout<<"1111111111111111100"<<endl;
                else
                        printf("%lld\n",ans[n]);
        }
        return 0;
}
```

E. RunningPhoton's Nightmare

题意

给出一个你n＊m的字符矩阵，.代表一块空白区域,S代表初始位置,E代表目标,W代表不能走到该区域,R代表这里有一个时间重置装置,可以重置炸弹的爆炸事件k。问从S到E的最小时间是多少?

分析

本来以为可以直接把结果搜出来，搜的状态是炸弹的时间还有实际走的时间，一直错。受原来做的一题的图论，分两步去搜索。本来是想去搜是否联通的，发现这样久丢掉了距离信息，于是乎第一次搜的时候就加上了距离的信息，这样就分两次做最短路就好了。

思考

不明白第一次搜的时候为什么会错?
注意模型的转化，如果不能直接求出来，转化一下，分两步求。

```
//  Created by Chenhongwei on 2016-05-07 Saturday 16:30
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
char s[610][610];
```

```
bool f[610][610];
int d[610][610];
int dp[610];
int dir[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
int n,m,k,cnt,ss,t;
bool check(int x,int y)
{
        if(x>=1&&x<=n&&m>=1&&y<=m&&s[x][y]!='W')
                return true;
        return false;
}
struct node
{
        int x,y;
        bool operator < (const node & rhs) const
        {
                if(x!=rhs.x)
                        return x<rhs.x;
                return y<rhs.y;
        }
}point[160];
vector<node> g[160];
void bfs(int x,int y)
{
        queue<node> q;
        node tmp,np;
        tmp.x=x,tmp.y=y;
        q.push(tmp);
        memset(f,0,sizeof f);
        memset(d,0x3f,sizeof d);
        d[x][y]=0;
        while(!q.empty())
        {
                tmp=q.front();
                q.pop();
                for(int i=0;i<4;i++)
                {
                        np.x=tmp.x+dir[i][0];
                        np.y=tmp.y+dir[i][1];
                        if(check(np.x,np.y)&&d[np.x][np.y]>d[tmp.x]
[tmp.y]+1)
                        {
```

```
                                d[np.x][np.y]=d[tmp.x][tmp.y]+1;
                                if(!f[np.x][np.y])
                                {
                                        f[np.x][np.y]=1;
                                        q.push(np);
                                }
                        }
                }
        }
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                cnt=0;
                node tmp;
                scanf("%d%d%d",&n,&m,&k);
                for(int i=1;i<=n;i++)
                {
                        scanf("%s",s[i]+1);
                        for(int j=1;j<=m;j++)
                                if(s[i][j]!='W'&&s[i][j]!='.')
                                {
                                        point[++cnt].x=i;
                                        point[cnt].y=j;
                                        if(s[i][j]=='S')
                                                ss=cnt;
                                        else if(s[i][j]=='E')
                                                t=cnt;
                                }
                }
                for(int i=0;i<=cnt;i++)
                        g[i].clear();
                for(int i=1;i<=cnt;i++)
                {
                        bfs(point[i].x,point[i].y);
                        for(int j=1;j<=cnt;j++)
```

```
                                if(i!=j)
                                {
                                        int xx=point[j].x,yy=point[j].y;
                                        if(d[xx][yy]<k)
                                        {
                                                g[i].push_back((node)
{j,d[xx][yy]});
                                                g[j].push_back((node)
{i,d[xx][yy]});
                                        }
                                }
                }
                memset(dp,0x3f,sizeof dp);

                dp[ss]=0;
                queue<int> q;
                bool flag[660];
                memset(flag,0,sizeof flag);
                flag[ss]=1;
                q.push(ss);
                while(!q.empty())
                {
                        int u=q.front();
                        q.pop();
                        for(int i=0;i<g[u].size();i++)
                        {
                                int v=g[u][i].x;
                                if(dp[v]>dp[u]+g[u][i].y)
                                {
                                        dp[v]=dp[u]+g[u][i].y;
                                        if(!flag[v])
                                        {
                                                q.push(v);
                                                flag[v]=1;
                                        }
                                }
                        }
                }
                if(dp[t]>inf)
                        puts("Poor RunningPhoton!");
                else
                        printf("%d\n",dp[t]);
```

```
        }
        return 0;
}
```

F. 表达式

题意

输入一个n(n<=1000),有一个x，最少需要几次乘除可以算出x^n

分析

暑假的时候做过的IDA*的题目，现实贪心，然后就是迭代加深搜索＋剪枝。

思考

虽然一眼就可以看出来，但是正确的剪枝姿势还是很重要的

```
//  Created by Chenhongwei on 2016-05-05 Thursday 23:20
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
```

```
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
int n,num,ans,dp[1010];
bool dfs(int cur,int d)
{
        if(d>=ans)
        {
                if(cur==n)
                        return true;
                return false;
        }
        int maxn=0;
        for(int j=0;j<num;j++)
                maxn=max(dp[j],maxn);
        if( (cur+maxn) << (ans-d-1) < n)
                return false;
        for(int i=num-1;i>=0;i--)
        {
                dp[num++]=cur+dp[i];
                if(dfs(cur+dp[i],d+1))
                        return true;
                num--;
                dp[num++]=cur-dp[i];
                if(dfs(cur-dp[i],d+1))
                        return true;
                num--;
        }
        return false;
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%d",&n);
                int tmp=1;
                ans=0;
```

```
                while(n>tmp)
                {
                        tmp*=2;
                        ans++;
                }
                while(1)
                {
                        memset(dp,0,sizeof dp);
                        dp[0]=num=1;
                        if(dfs(1,0))break;
                        ans++;
                }
                printf("%d\n",ans);
        }
        return 0;
}
```

G. 神舟的宝藏

题意

找出一个最小的N的倍数，这个数字是C进制，且每个数位上只能使用给定的N个数字

分析

肯定不能直接暴力的去枚举。如果一个数%N==0，那么这个数就是N的倍数。在没有找到的前提下，如果A%N==B%N，而且A<B，那么其实我们就可以取A而不取B，因为如果在A末尾增加C可以使得AC%N==0，那么BC%N也等于0，易得：如果A和B追加数之后%N==0，那么最优条件下追加的数肯定相同。这样，直接取余判重就好啦，因为N范围很小，这样就使得去搜的状态大大减少

思考

主要还是取余判重的思想

```
//  Created by Chenhongwei on 2016-05-07 Saturday 21:09
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
```

```
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
bool num[20],vis[5500];
int n,c,m;
struct node
{
        int len;
        int s[550];
};
int mod(node p)
{
        int ret=0;
        for(int i=0;i<p.len;i++)
                ret=(ret*c+p.s[i])%n;
        return ret;
}
void print(node p)
{
        for(int i=0;i<p.len;i++)
                if(p.s[i]<10)
                        printf("%d",p.s[i]);
                else
                        printf("%c",'A'+(p.s[i]-10));
        printf("\n");
}
```

```
bool bfs()
{
        memset(vis,0,sizeof vis);
        queue<node> q;
        node p;
        p.len=0;
        int r;
        for(int i=1;i<16;i++)
                if(num[i])
                {
                        p.len=1;
                        p.s[0]=i;
                        r=mod(p);
                        if(!r)
                        {
                                print(p);
                                return 1;
                        }
                        else if(!vis[r])
                        {
                                vis[r]=1;
                                q.push(p);
                        }
                }
        while(!q.empty())
        {
                p=q.front();
                q.pop();
                for(int i=0;i<16;i++)
                        if(num[i])
                        {
                                p.s[p.len]=i;
                                p.len++;
                                r=mod(p);
                                if(!r)
                                {
                                        print(p);
                                        return 1;
                                }
                                else if(!vis[r]&&p.len<499)
                                {
                                        vis[r]=1;
```

```
                                        q.push(p);
                                }
                                p.len--;
                        }
        }
        return 0;
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        char str[5];
        while(T--)
        {
                scanf("%d%d%d",&n,&c,&m);
                memset(num,0,sizeof num);
                for(int i=1;i<=m;i++)
                {
                        scanf("%s",str);
                        if(str[0]>='0'&&str[0]<='9')
                                num[str[0]-'0']=1;
                        else
                                num[str[0]-'A'+10]=1;
                }
                if(n)
                {
                        bool flag=bfs();
                        if(!flag)
                                puts("BOMB!!!");
                }
                else
                {
                        if(num[0])
                                printf("0\n");
                        else
                                puts("BOMB!!!");
                }
        }
        return 0;
```

```
}
```

H. DNA序列

题意

找出一个最短的DAN序列，使得它按图中给的方式包含所有给出的DNA序列

分析

用IDA*的方法去搜索，去搜的时候判断每走一步要走的有意义，就是当添加的字符至少与一个给定的DNA序列串未被匹配的字符重合

思考

注意剪枝

```
//  Created by Chenhongwei on 2016-05-09 Monday 18:10
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e5+100;
```

```
typedef long long ll;
typedef unsigned long long ull;
int n,m,ans;
char s[10][10];
char D[4]={'A','C','G','T'};
int len[10];
void dfs(int *p,int d,int dep)
{
        if(d>dep||ans!=-1)
                return ;
        int maxv=0;
        for(int i=1;i<=n;i++)
        {
                int t=len[i]-p[i];
                maxv=max(maxv,t);
        }
        if(maxv==0)
        {
                ans=d;
                return ;
        }
        if(d+maxv>dep)
                return ;
        for(int i=0;i<4;i++)
        {
                int q[10];
                bool flag=0;
                for(int j=1;j<=n;j++)
                        if(s[j][p[j]]==D[i])
                        {
                                flag=1;
                                q[j]=p[j]+1;
                        }
                        else
                                q[j]=p[j];
                if(flag)
                        dfs(q,d+1,dep);
                if(ans!=-1)
                        break;
        }
}
int main()
```

```
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%d",&n);
                int maxlen=0;
                for(int i=1;i<=n;i++)
                {
                        scanf("%s",s[i]);
                        len[i]=strlen(s[i]);
                        maxlen=max(maxlen,len[i]);
                }
                int dep=maxlen;
                ans=-1;
                int p[20];
                memset(p,0,sizeof p);
                while(true)
                {
                        dfs(p,0,dep);
                        if(ans!=-1)
                                break;
                        dep++;
                }
                printf("%d\n",ans);
        }
        return 0;
}
```

I. 小冰和小娜

题意

给出一个你n*m的字符矩阵，#代表障碍物，.代表空地，S代表起点，T代表终点，左转，右转和直走，并且每一步都会花费等量的时间，转向不会引起接触点的颜色改变,直走会使颜色变成下一个颜色，问能不能从S走到E且颜色不改变

分析

需要考虑的状态就比较多了，首先是坐标，然后是颜色，还有方向。一开始看漏了条件，没有注意到颜色，wa了1次

思考

注意走的时候的方向和颜色的判断就好了，写的比较复杂

```
//  Created by Chenhongwei on 2016-05-06 Friday 18:52
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
char s[55][55];
int dp[55][55][5][6];
bool f[55][55][5][6];
struct node
{
        int x,y,dir,c;//1 up 2 left 3 down 4 right
};
int m,n;
bool check(int x,int y)
```

```
{
        if(x>=1&&x<=m&&y>=1&&y<=n&&s[x][y]!='#')
                return true;
        return false;
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                int sx,sy,ex,ey;
                scanf("%d%d",&m,&n);
                for(int i=1;i<=m;i++)
                {
                        scanf("%s",s[i]+1);
                        for(int j=1;j<=n;j++)
                                if(s[i][j]=='S')
                                        sx=i,sy=j;
                                else if(s[i][j]=='T')
                                        ex=i,ey=j;
                }
                memset(dp,0x3f,sizeof dp);
                memset(f,0,sizeof f);
                dp[sx][sy][1][0]=0;
                f[sx][sy][1][0]=1;
                queue<node> q;
                node p,np;
                p.x=sx,p.y=sy,p.dir=1,p.c=0;
                q.push(p);
                while(!q.empty())
                {
                        p=q.front();
                        f[p.x][p.y][p.dir][p.c]=0;
                        // cout<<p.x<<' '<<p.y<<' '<<p.dir<<endl;
                        q.pop();
                        if(p.dir==1)
                        {
                                if(check(p.x-1,p.y)&&dp[p.x-1][p.y][1]
```

```
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+1)
                                {
                                        dp[p.x-1][p.y][1]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+1;
                                        if(!f[p.x-1][p.y][1][(p.c+1)%5])
                                        {
                                                np.x=p.x-
1,np.y=p.y,np.dir=1,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y-1)&&dp[p.x][p.y-1][2]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x][p.y-1][2]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x][p.y-1][2][(p.c+1)%5])
                                        {
                                                np.x=p.x,np.y=p.y-
1,np.dir=2,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x+1,p.y)&&dp[p.x+1][p.y][3]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+3)
                                {
                                        dp[p.x+1][p.y][3]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+3;
                                        if(!f[p.x+1][p.y][3][(p.c+1)%5])
                                        {

np.x=p.x+1,np.y=p.y,np.dir=3,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y+1)&&dp[p.x][p.y+1][4]
```

```
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x][p.y+1][4]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x][p.y+1][4][(p.c+1)%5])
                                        {

np.x=p.x,np.y=p.y+1,np.dir=4,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                        }
                        else if(p.dir==2)
                        {
                                if(check(p.x-1,p.y)&&dp[p.x-1][p.y][1]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x-1][p.y][1]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x-1][p.y][1][(p.c+1)%5])
                                        {
                                                np.x=p.x-
1,np.y=p.y,np.dir=1,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y-1)&&dp[p.x][p.y-1][2]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+1)
                                {
                                        dp[p.x][p.y-1][2]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+1;
                                        if(!f[p.x][p.y-1][2][(p.c+1)%5])
                                        {
                                                np.x=p.x,np.y=p.y-
1,np.dir=2,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
```

```
                                        }
                                }
                                if(check(p.x+1,p.y)&&dp[p.x+1][p.y][3]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x+1][p.y][3]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x+1][p.y][3][(p.c+1)%5])
                                        {

np.x=p.x+1,np.y=p.y,np.dir=3,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y+1)&&dp[p.x][p.y+1][4]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+3)
                                {
                                        dp[p.x][p.y+1][4]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+3;
                                        if(!f[p.x][p.y+1][4][(p.c+1)%5])
                                        {

np.x=p.x,np.y=p.y+1,np.dir=4,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                        }
                        else if(p.dir==3)
                        {
                                if(check(p.x-1,p.y)&&dp[p.x-1][p.y][1]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+3)
                                {
                                        dp[p.x-1][p.y][1]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+3;
                                        if(!f[p.x-1][p.y][1][(p.c+1)%5])
                                        {
                                                np.x=p.x-
1,np.y=p.y,np.dir=1,np.c=(p.c+1)%5;
```

```
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y-1)&&dp[p.x][p.y-1][2]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x][p.y-1][2]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x][p.y-1][2][(p.c+1)%5])
                                        {
                                                np.x=p.x,np.y=p.y-
1,np.dir=2,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x+1,p.y)&&dp[p.x+1][p.y][3]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+1)
                                {
                                        dp[p.x+1][p.y][3]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+1;
                                        if(!f[p.x+1][p.y][3][(p.c+1)%5])
                                        {

np.x=p.x+1,np.y=p.y,np.dir=3,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y+1)&&dp[p.x][p.y+1][4]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x][p.y+1][4]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x][p.y+1][4][(p.c+1)%5])
                                        {

np.x=p.x,np.y=p.y+1,np.dir=4,np.c=(p.c+1)%5;
```

```
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                        }
                        else if(p.dir==4)
                        {
                                if(check(p.x-1,p.y)&&dp[p.x-1][p.y][1]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x-1][p.y][1]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x-1][p.y][1][(p.c+1)%5])
                                        {
                                                np.x=p.x-
1,np.y=p.y,np.dir=1,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y-1)&&dp[p.x][p.y-1][2]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+3)
                                {
                                        dp[p.x][p.y-1][2]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+3;
                                        if(!f[p.x][p.y-1][2][(p.c+1)%5])
                                        {
                                                np.x=p.x,np.y=p.y-
1,np.dir=2,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x+1,p.y)&&dp[p.x+1][p.y][3]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+2)
                                {
                                        dp[p.x+1][p.y][3]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+2;
                                        if(!f[p.x+1][p.y][3][(p.c+1)%5])
```

```
                                        {

np.x=p.x+1,np.y=p.y,np.dir=3,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                                if(check(p.x,p.y+1)&&dp[p.x][p.y+1][4]
[(p.c+1)%5]>dp[p.x][p.y][p.dir][p.c]+1)
                                {
                                        dp[p.x][p.y+1][4]
[(p.c+1)%5]=dp[p.x][p.y][p.dir][p.c]+1;
                                        if(!f[p.x][p.y+1][4][(p.c+1)%5])
                                        {

np.x=p.x,np.y=p.y+1,np.dir=4,np.c=(p.c+1)%5;
                                                f[np.x][np.y][np.dir]
[np.c]=1;
                                                q.push(np);
                                        }
                                }
                        }

                }
                int ans=dp[ex][ey][1][0];
                for(int i=2;i<=4;i++)
                        ans=min(ans,dp[ex][ey][i][0]);
                if(ans>1e9)
                        printf("-1\n");
                else
                        printf("%d\n",ans);
        }
        return 0;
}
```

J. TooEasy Or TooDifficult

题意

强行拼凑起来的题,第一个实际上就是以每个字符为中心的回文串的长度，第二问就是給n个数字，找出一个区间，是这个区间的异或值最大。

分析

都是老题，抢行拼在一起。第一问就是manacher算法＋快速幂。第二个就是用字典树把前n个数的异或值插进去，然后贪心找最大值

思考

注意数据范围，要用long long

```
//  Created by Chenhongwei on 2016-05-09 Monday 17:08
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
char s[maxn],t[maxn*2];
ll p[maxn*2],pp[maxn*2];
ll n,m,cnt;
ll mz,mod;
ll a[maxn],v[maxn*34];
```

```
ll nxt[maxn*34][2];
void init()
{
        t[0]='$',t[1]='#';
        for(ll i=0;i<n;i++)
        {
                t[i*2+2]=s[i];
                t[i*2+3]='#';
        }
        n=n*2+2;
        t[n]=0;
}
void manacher()
{
        ll maxv=0,pos;
        for(ll i=1;i<n;i++)
        {
                if(maxv>i)
                        p[i]=min(p[2*pos-i],maxv-i);
                else
                        p[i]=1;
                while(t[i-p[i]]==t[i+p[i]]&&i+p[i]<n)
                        p[i]++;
                if(p[i]+i>maxv)
                {
                        maxv=p[i]+i;
                        pos=i;
                }
        }
}
void insert(int x)
{
        ll u=0;
        for(ll i=31;i>=0;i--)
        {
                ll k=(x>>i)&1;
                if(!nxt[u][k])
                {
                        memset(nxt[cnt],0,sizeof nxt[cnt]);
                        v[cnt]=0;
                        nxt[u][k]=cnt++;
                }
```

```
                u=nxt[u][k];
                v[u]++;
        }
}
ll query(int x)
{
        ll u=0;
        ll ret=0;
        for(ll i=31;i>=0;i--)
        {
                ll k=((x>>i)&1)^1;
                if(nxt[u][k])
                        ret|=(1<<i);
                else
                        k^=1;
                u=nxt[u][k];
        }
        return ret;
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%s",s);
                m=n=strlen(s);
                init();
                manacher();
                mz=1;
                ll tt=0;
                for(ll i=0;i<n;i++)
                        if(t[i]!='#'&&t[i]!='$')
                        {
                                mz=max(mz,p[i]-1);
                                pp[++tt]=p[i]-1;
                        }
                ll num=m*m*m;
                ll mod=mz/3*5+1;
```

```
                ll JD=1,tmp=mz;
                while(num)
                {
                        if(num&1)
                                JD=JD*tmp%mod;
                        tmp=(tmp*tmp)%mod;
                        num>>=1;
                }
                JD+=(mz*4/5);
                cnt=1,v[0]=0;
                memset(nxt[0],0,sizeof nxt[0]);
                for(ll i=1;i<=tt;i++)
                {
                        a[i]=a[i-1]^pp[i];
                        insert(a[i]);
                }
                ll FJD=-1;
                for(ll i=0;i<=tt;i++)
                        FJD=max(FJD,query(a[i]));
                if(JD>FJD)
                        puts("liujc");
                else
                        puts("luoxinchen");
        }
        return 0;
}
```

K. 奶牛合影

题意

对于一个可以按顺时针顺序旋转的数组，问什么时候他的字典序最小

分析

最小表示法裸题,好像原来的模版有问题

思考

```
//  Created by Chenhongwei on 2016-05-06 Friday 17:12
//  Copyright (c) 2016 Chenhongwei. All rights reserved.
```

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=3e5+100;
typedef long long ll;
typedef unsigned long long ull;
int a[maxn],n;
int minrepresent(){
    int i=0,j=1,k=0,t;
    while(i<n && j<n && k<n){
        t = a[(i+k) >= n ? i+k-n : i+k] - a[(j+k) >= n ? j+k-n : j+k];
        if(!t)k++;
        else {
            if(t>0)i = i+k+1;
            else j = j+k+1;
            if(i==j)j++;
            k = 0;
        }
    }
    return i;
}
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
```

```
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        while(T--)
        {
                scanf("%d",&n);
                for(int i=0;i<n;i++)
                        scanf("%d",&a[i]);
                int p=minrepresent();
                printf("%d",a[p]);
                for(int i=p+1;i<n;i++)
                        printf(" %d",a[i]);
                for(int i=0;i<p;i++)
                        printf(" %d",a[i]);
                printf("\n");
        }
        return 0;
}
```

G. 奶牛硬盘

题意

计算硬盘标示和系统显示的差值的百分比

分析

直接按照计算方式的差异的百分比去算就好了

思考

```
//  Created by Chenhongwei on 2016-05-06 Friday 15:30
//  Copyright (c) 2016 Chenhongwei. All rights reserved.

#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <climits>
#include <queue>
#include <cmath>
```

```
#include <map>
#include <set>
#include <stack>
#include <vector>
#include <sstream>
#include <algorithm>
#define root 1,n,1
#define lson l,m,rt<<1
#define rson m+1,r,rt<<1|1
using namespace std;
const int inf=1e9;
const int mod=1e9+7;
const int maxn=1e5+100;
typedef long long ll;
typedef unsigned long long ull;
int main()
{
        //ios::sync_with_stdio(false);
    // freopen("in.txt","r",stdin);
        //freopen("out.txt","w",stdout);
        int T;
        scanf("%d",&T);
        for(int kase=1;kase<=T;kase++)
        {
                int tmp;
                char s[20];
                scanf("%d",&tmp);
                scanf("%s",s);
                if(s[1]=='B')
                        tmp=0;
                else if(s[1]=='K')
                        tmp=1;
                else if(s[1]=='M')
                        tmp=2;
                else if(s[1]=='G')
                        tmp=3;
                else if(s[1]=='T')
                        tmp=4;
                else if(s[1]=='P')
                        tmp=5;
                else if(s[1]=='E')
                        tmp=6;
```

```
                else if(s[1]=='Z')
                        tmp=7;
                else if(s[1]=='Y')
                        tmp=8;
                double C=(double)1000/(double)1024;
                double ans=1.0;
                for(int i=1;i<=tmp;i++)
                        ans=ans*C;
                ans=1-ans;
                ans*=100;
                printf("Case #%d: %.2lf%%\n",kase,ans);

        }
        return 0;
}
```

L. 奶牛序列

题意

一个长为n的字符串S，计算所给的公式的值

分析

想的用扩展kmp去做，发现复杂度降不下来。问别人才知道用后缀数组可以直接求，用的被人的代码。

思考