初赛题解

重庆大学第十五届程序设计大赛

A.数量多少?

12个，数一下就好，欢迎加入ACM社团

B.上下左右?

仔细观察一下就能发现，只要找到箭头的拐点，然后判断出两条斜边的方向，就能知道是什么箭头了

```
/*
 * Author:  JiangYu
 * Created Time:  2018/5/17 17:50:42
 * File Name: B.cpp
 */
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <cmath>
#include <algorithm>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <time.h>
using namespace std;
#define ll long long
#define MP make_pair
#define PB push_back
#define X first
#define Y second
#define FI first
#define SE second
#define inf 0x3f3f3f3f
#define FOR(i,a,b) for(int i = a; i <= b; ++i)
#define FORD(i,a,b) for(int i = b; i >= a; --i)
#define ALL(x) x.begin(),x.end()
#define REP(i,a) for(int i = 0; i < a; ++i)
#define DEP(i,a) for(int i = a-1; i >= 0; --i)
#define CLR(a) memset(a, 0, sizeof a)
char g[20][20];
int main() {
    for(int i = 1; i <= 10; ++i) {
        scanf("%s", g[i]+1);
    }
    for(int i = 1; i <= 10; ++i) {
        for(int j = 1; j <= 10; ++j) {
            if(g[i][j] == '#' && g[i+1][j-1] == '#' && g[i+1][j+1] == '#') {
                printf("UP\n");
                return 0;
            }
            else if(g[i][j] == '#' && g[i+1][j-1] == '#' && g[i-1][j-1] == '#') {
                printf("RIGHT\n");
```

```
                return 0;
            }
            else if(g[i][j] == '#' && g[i+1][j+1] == '#' && g[i-1][j+1] == '#') {
                printf("LEFT\n");
                return 0;
            }
            else if(g[i][j] == '#' && g[i-1][j-1] == '#' && g[i-1][j+1] == '#') {
                printf("DOWN\n");
                return 0;
            }
        }
    }
    return 0;
}
```

C.高数过没?

暴力计算显然不行，复杂度为$O(tp)$
注意到a很少，所以可以预处理出所有的a^p,复杂度为$O(t+p)$
当然时间给的很足，也可以用快速幂暴力计算
（预处理解法）

```
/*
 * Author:  JiangYu
 * Created Time:  2018/5/16 23:35:53
 * File Name: D.cpp
 */
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define MP make_pair
#define PB push_back
#define X first
#define Y second
#define FI first
#define SE second
#define inf 0x3f3f3f3f
#define FOR(i,a,b) for(int i = a; i <= b; ++i)
#define FORD(i,a,b) for(int i = b; i >= a; --i)
#define ALL(x) x.begin(),x.end()
#define REP(i,a) for(int i = 0; i < a; ++i)
#define DEP(i,a) for(int i = a-1; i >= 0; --i)
#define CLR(a) memset(a, 0, sizeof a)
const int N = 1e7 + 1024;
ll ans[2][N];
const ll mod = 1e9 + 7;
int main() {
    ans[0][0] = ans[1][0] = 1;
    for(int i = 1; i < N; ++i) {
        for(int j = 2; j <= 3; ++j) {
            ans[j - 2][i] = 111 * ans[j - 2][i - 1] * j % mod;
        }
    }
    int t;
    scanf("%d", &t);
    while(t--) {
        int a, p;
        scanf("%d%d", &a, &p);
        if(a == 0) puts("0");
        else if(a == 1) puts("1");
        else {
            printf("%d\n", ans[a-2][p]);
        }

```

```
    }
    return 0;
}

```

（快速幂解法）

```
/*
 * Author:  JiangYu
 * Created Time:  2018/5/16 23:25:09
 * File Name: C.cpp
 */
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define MP make_pair
#define PB push_back
#define X first
#define Y second
#define FI first
#define SE second
#define inf 0x3f3f3f3f
#define FOR(i,a,b) for(int i = a; i <= b; ++i)
#define FORD(i,a,b) for(int i = b; i >= a; --i)
#define ALL(x) x.begin(),x.end()
#define REP(i,a) for(int i = 0; i < a; ++i)
#define DEP(i,a) for(int i = a-1; i >= 0; --i)
#define CLR(a) memset(a, 0, sizeof a)

const ll mod = 1e9 + 7;
ll po(ll a, ll p) {
    ll base = 1;
    a %= mod;
    while(p) {
        if(p & 1) base = base * a % mod;
        a = a * a % mod;
        p >>= 1;
    }
    return base % mod;
}
int main() {
    int t;
    scanf("%d", &t);
    while(t--) {
        ll a, p;
        scanf("%lld%lld", &a, &p);
        if(a == 0) puts("0");
        else if(a == 1) puts("1");
        else {
            printf("%lld\n", po(a, p));
        }
    }
    return 0;
}


```

D.还是数学?

我们把$A > B$视为一条从A连向B的边，同理$A < B$视为一条从B连向A的边。则题意转换为n个点，m条边的有向图中，是否存在至少一个环?
然后这道题就可以通过简单的搜索解决了。
vis[i] == 0 表示 i 点还未访问过

vis[i] == -1 表示 i 点已经访问过，但是从i点出发的点还未访问完
vis[i] == 1 表示 i 点已经访问过，并且从i点出发的点全部访问完毕
那么在搜索时，如果遇到从u点出发到达v点时，vis[v] == -1, 则表示u是一个从v出发到达的点，同时u还能到达v -> 即u，v之间存在一个环。

```
/*
 * Author:  JiangYu
 * Created Time:  2018/5/17 16:10:13
 * File Name: D.cpp
 */
#pragma comment(linker, "/STACK:102400000,102400000")
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define MP make_pair
#define PB push_back
#define X first
#define Y second
#define FI first
#define SE second
#define inf 0x3f3f3f3f
#define FOR(i,a,b) for(int i = a; i <= b; ++i)
#define FORD(i,a,b) for(int i = b; i >= a; --i)
#define ALL(x) x.begin(),x.end()
#define REP(i,a) for(int i = 0; i < a; ++i)
#define DEP(i,a) for(int i = a-1; i >= 0; --i)
#define CLR(a) memset(a, 0, sizeof a)
const int N = 1e5 + 1024;
struct edge{
    int to, nex;
}ed[N];
int head[N] , cnt;
int vis[N];
void add(int from, int to) {
    ed[cnt].to = to;
    ed[cnt].nex = head[from];
    head[from] = cnt++;
}
void dfs(int u) {
    vis[u] = -1;
    for(int i = head[u]; ~i; i = ed[i].nex) {
        int v = ed[i].to;
        if(vis[v] == 0) {
           dfs(v);
           vis[v] = 1;
        }
        if(vis[v] == -1) {
            printf("ohu!");
            exit(0);
        }
    }
}
int main() {
    memset(head, -1, sizeof head);
    memset(vis, 0, sizeof vis);
    int n, m;
    scanf("%d%d", &n, &m);
    for(int i = 1; i <= m; ++i) {
        int u, v;
        char ch[5];
        scanf("%d%s%d", &u, ch, &v);
        if(ch[0] == '>') {
            add(u, v);
        }
        else {
            add(v, u);
        }
```

```
        }
        for(int i = 1; i <= n; ++i) {
            if(vis[i] == 0) {
                dfs(i);
                vis[i] = 1;
            }
        }
        printf("yeah~");
        return 0;
    }

```

+