

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

```
1.  /*
2.   * Author:   JiangYu
3.   * Created Time: 2018/5/17 17:50:42
4.   * File Name: B.cpp
5.   */
6.  #include <iostream>
7.  #include <cstdio>
8.  #include <cstdlib>
9.  #include <cstring>
10. #include <cmath>
11. #include <algorithm>
12. #include <string>
13. #include <vector>
14. #include <stack>
15. #include <queue>
16. #include <set>
17. #include <map>
18. #include <time.h>
19. using namespace std;
20. #define ll long long
21. #define MP make_pair
22. #define PB push_back
23. #define X first
24. #define Y second
25. #define FI first
26. #define SE second
27. #define inf 0x3f3f3f3f
28. #define FOR(i,a,b) for(int i = a; i <= b; ++i)
29. #define FORD(i,a,b) for(int i = b; i >= a; --i)
30. #define ALL(x) x.begin(), x.end()
31. #define REP(i,a) for(int i = 0; i < a; ++i)
32. #define DEP(i,a) for(int i = a-1; i >= 0; --i)
33. #define CLR(a) memset(a, 0, sizeof a)
34. char g[20][20];
35. int main() {
36.     for(int i = 1; i <= 10; ++i) {
37.         scanf("%s", g[i]+1);
38.     }
39.     for(int i = 1; i <= 10; ++i) {
40.         for(int j = 1; j <= 10; ++j) {
41.             if(g[i][j] == '#' && g[i+1][j-1] == '#' && g[i+1][j+1] == '#') {
42.                 printf("UP\n");
43.                 return 0;
44.             }
45.             else if(g[i][j] == '#' && g[i+1][j-1] == '#' && g[i-1][j-1] == '#') {
46.                 printf("RIGHT\n");
```

```

47.         return 0;
48.     }
49.     else if(g[i][j] == '#' && g[i+1][j+1] == '#' && g[i-1][j+1] == '#') {
50.         printf("LEFT\n");
51.         return 0;
52.     }
53.     else if(g[i][j] == '#' && g[i-1][j-1] == '#' && g[i-1][j+1] == '#') {
54.         printf("DOWN\n");
55.         return 0;
56.     }
57. }
58. }
59. return 0;
60. }

```

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)

```

1.  /*
2.   * Author:   JiangYu
3.   * Created Time: 2018/5/16 23:35:53
4.   * File Name: D. cpp
5.   */
6.  #include <bits/stdc++.h>
7.  using namespace std;
8.  #define ll long long
9.  #define MP make_pair
10. #define PB push_back
11. #define X first
12. #define Y second
13. #define FI first
14. #define SE second
15. #define inf 0x3f3f3f3f
16. #define FOR(i,a,b) for(int i = a; i <= b; ++i)
17. #define FORD(i,a,b) for(int i = b; i >= a; --i)
18. #define ALL(x) x.begin(),x.end()
19. #define REP(i,a) for(int i = 0; i < a; ++i)
20. #define DEP(i,a) for(int i = a-1; i >= 0; --i)
21. #define CLR(a) memset(a, 0, sizeof a)
22. const int N = 1e7 + 1024;
23. ll ans[2][N];
24. const ll mod = 1e9 + 7;
25. int main() {
26.     ans[0][0] = ans[1][0] = 1;
27.     for(int i = 1; i < N; ++i) {
28.         for(int j = 2; j <= 3; ++j) {
29.             ans[j-2][i] = 111 * ans[j-2][i-1] * j % mod;
30.         }
31.     }
32.     int t;
33.     scanf("%d", &t);
34.     while(t--) {
35.         int a, p;
36.         scanf("%d%d", &a, &p);
37.         if(a == 0) puts("0");
38.         else if(a == 1) puts("1");
39.         else {
40.             printf("%d\n", ans[a-2][p]);
41.         }
42.     }

```

```

43.     }
44.     return 0;
45. }
46.

```

(快速幂解法)

```

1.  /*
2.   * Author:   JiangYu
3.   * Created Time: 2018/5/16 23:25:09
4.   * File Name: C.cpp
5.   */
6.  #include <bits/stdc++.h>
7.  using namespace std;
8.  #define ll long long
9.  #define MP make_pair
10. #define PB push_back
11. #define X first
12. #define Y second
13. #define FI first
14. #define SE second
15. #define inf 0x3f3f3f3f
16. #define FOR(i,a,b) for(int i = a; i <= b; ++i)
17. #define FORD(i,a,b) for(int i = b; i >= a; --i)
18. #define ALL(x) x.begin(), x.end()
19. #define REP(i,a) for(int i = 0; i < a; ++i)
20. #define DEP(i,a) for(int i = a-1; i >= 0; --i)
21. #define CLR(a) memset(a, 0, sizeof a)
22.
23. const ll mod = 1e9 + 7;
24. ll po(ll a, ll p) {
25.     ll base = 1;
26.     a %= mod;
27.     while(p) {
28.         if(p & 1) base = base * a % mod;
29.         a = a * a % mod;
30.         p >>= 1;
31.     }
32.     return base % mod;
33. }
34. int main() {
35.     int t;
36.     scanf("%d", &t);
37.     while(t--) {
38.         ll a, p;
39.         scanf("%lld%lld", &a, &p);
40.         if(a == 0) puts("0");
41.         else if(a == 1) puts("1");
42.         else {
43.             printf("%lld\n", po(a, p));
44.         }
45.     }
46.     return 0;
47. }
48.
49.

```

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。

```
1.  /*
2.   * Author:   JiangYu
3.   * Created Time: 2018/5/17 16:10:13
4.   * File Name: D.cpp
5.   */
6.  #pragma comment(linker, "/STACK:102400000,102400000")
7.  #include <bits/stdc++.h>
8.  using namespace std;
9.  #define ll long long
10. #define MP make_pair
11. #define PB push_back
12. #define X first
13. #define Y second
14. #define FI first
15. #define SE second
16. #define inf 0x3f3f3f3f
17. #define FOR(i,a,b) for(int i = a; i <= b; ++i)
18. #define FORD(i,a,b) for(int i = b; i >= a; --i)
19. #define ALL(x) x.begin(),x.end()
20. #define REP(i,a) for(int i = 0; i < a; ++i)
21. #define DEP(i,a) for(int i = a-1; i >= 0; --i)
22. #define CLR(a) memset(a, 0, sizeof a)
23. const int N = 1e5 + 1024;
24. struct edge{
25.     int to, nex;
26. }ed[N];
27. int head[N] , cnt;
28. int vis[N];
29. void add(int from, int to) {
30.     ed[cnt].to = to;
31.     ed[cnt].nex = head[from];
32.     head[from] = cnt++;
33. }
34. void dfs(int u) {
35.     vis[u] = -1;
36.     for(int i = head[u]; ~i; i = ed[i].nex) {
37.         int v = ed[i].to;
38.         if(vis[v] == 0) {
39.             dfs(v);
40.             vis[v] = 1;
41.         }
42.         if(vis[v] == -1) {
43.             printf("ohu!");
44.             exit(0);
45.         }
46.     }
47. }
48. int main() {
49.     memset(head, -1, sizeof head);
50.     memset(vis, 0, sizeof vis);
51.     int n, m;
52.     scanf("%d%d", &n, &m);
53.     for(int i = 1; i <= m; ++i) {
54.         int u, v;
55.         char ch[5];
56.         scanf("%d%s%d", &u, ch, &v);
57.         if(ch[0] == '>') {
58.             add(u, v);
59.         }
60.         else {
61.             add(v, u);
62.         }
63.     }
```

```
63.     }
64.     for(int i = 1; i <= n; ++i) {
65.         if(vis[i] == 0) {
66.             dfs(i);
67.             vis[i] = 1;
68.         }
69.     }
70.     printf("yeah~");
71.     return 0;
72. }
73.
```

+