Тренинг-центр «Профит»
г. Красноярск, ул. Авиаторов, 44
http://vk.com/profit_krsk

Разбор задач олимпиады «Четвёртая личная олимпиада» сайта acmp.ru от 20.11.2013 года

Задача A. Забавная последовательность

Для решения этой задачи надо быстро уметь проверять встречалось ли число в последовательности. В общем случае для этой цели надо использовать множества, но заметим, что при данных ограничениях все числа последовательности будут не очень большими. Действительно, даже если бы на каждом шаге происходило прибавление тройки, то стотысячное число было бы равно 300001. Поэтому воспользуемся массивом, в i -ом элементе которого будем хранить единичку, если число уже встречалось.

Решение на языке C++

```cpp
#include <iostream>
#include <cstdio>
#include <memory.h>

using namespace std;

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    int n, a=1, m[300500];
    memset(m, 0, sizeof(m));
    m[1] = 1;
    cin>>n;
    for(int i=2; i<=n; ++i){
        if(m[i]) a += 3;
        else a += 2;
        m[a] = 1;
    }
    cout<<a;
    return 0;
}
```

Решение на языке Паскаль

```pascal
var
    n, a, i : Longint;
    m : array [1..300500] of Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);
    m[1] := 1;
    a := 1;
    Read(n);
    for i:=2 to n do begin
        if m[i] = 1 then inc(a, 3)
        else inc(a, 2);
        m[a] := 1;
    end;
    Write(a);
end.
```

Задача B. Треугольник - 4

Очевидно, что координаты вершин требуемого треугольника отклоняются от заданной точки не менее чем на 1 в каждом из четырёх направлений. Это легко доказать от противного, предположив, что ни одна из трёх вершин не имеет координату больше x. Тогда получим, что все вершины расположены на одной прямой. А такой треугольник имеет нулевую площадь и не подходит.
Заметим, что треугольник вида:

Удовлетворяет всем условиям задачи, а координаты его вершин отклоняются от точки (x, y) ровно на 1. Поэтому можно считать это оптимальным вариантом. В таком случае решение задачи становится очень простым.

Решение на языке C++

```cpp
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    int x, y;
    cin>>x>>y;
    if(abs(x) < 1e9 && abs(y) < 1e9)
        cout<<"YES\n"<<x+1<<" "<<y+1<<endl<<x<<" "<<y-1<<endl<<x-1<<" "<<y;
    else
        cout<<"NO";
    return 0;
}
```

Решение на языке Паскаль

```pascal
var
    x, y : Longint;
begin
    assign(input, 'input.txt'); reset(input);
    assign(output, 'output.txt'); rewrite(output);
    Read(x, y);
    if (abs(x) < 1e9) and (abs(y) < 1e9) then begin
        WriteLn('YES');
        WriteLn(x+1,' ',y+1);
        WriteLn(x,' ',y-1);
        WriteLn(x-1,' ',y);
    end else Write('NO');
end.
```

Задача C. ePig

В данной задаче необходимо понять, что количество раундов не будет большим. Поэтому можно промоделировать действия, описанные в условии.
Приведённое решение не содержит никаких сложным моментов, поэтому поясним лишь назначение массивов:

mm – логический массив, в котором отмечаем наличие куска у клиента,

mk – количество клиентов, у которых есть i -ый кусок,

mn – количество кусков у клиента i ,

rk – номер желаемого куска для клиента i ,
rn – номер клиента, у которого желает качать клиент i ,
qn – номер клиента, которому будет качать клиент i ,
mc – ценность клиентов,
res – на каком раунде клиент i закончил скачивание.

Решение на языке C++

```
#include <iostream>
#include <cstdio>
#include <memory.h>

using namespace std;

bool mm[100][200];
int mk[200], rk[100], mn[100], rn[100], qn[100], mc[100][100], res[100];

int main(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);

    int n, k;
    cin>>n>>k;

    for(int i=0;i<k;++i) mm[0][i] = true, mk[i] = 1;
    memset(mc, 0, sizeof(mc));
    memset(mn, 0, sizeof(mn));
    mn[0] = k;
    memset(res, -1, sizeof(res));

    int ck = (n-1) * k, it=0;
    while(ck){

        memset(rk, -1, sizeof(rk));
        for(int i=0;i<n;++i)
            for(int j=0;j<k;++j)
                if(!mm[i][j] && (rk[i]<0 || mk[rk[i]] > mk[j])) rk[i] = j;

        memset(rn, -1, sizeof(rn));
        for(int i=0;i<n;++i){
            if(rk[i] < 0) continue ;
            int t = rk[i];
            for(int j=0;j<n;++j)
                if(mm[j][t] && (rn[i]<0 || mn[rn[i]] > mn[j])) rn[i] = j;
        }

        memset(qn, -1, sizeof(qn));
        for(int i=0;i<n;++i){
            if(rn[i] < 0) continue ;
            int t = rn[i];
            if(qn[t] < 0 ||
               mc[t][qn[t]] < mc[t][i] ||
               (mc[t][qn[t]] == mc[t][i] && mn[qn[t]] > mn[i])) qn[t] = i;
        }

        for(int i=0;i<n;++i){
            if(qn[i] < 0) continue ;
            --ck;
            ++mc[qn[i]][i];
            ++mn[qn[i]];
            ++mk[rk[qn[i]]];
            mm[qn[i]][rk[qn[i]]] = true;
            if(mn[qn[i]] == k) res[qn[i]] = it+1;
        }
        ++it;
    }

    for(int i=1;i<n;++i) cout<<res[i]<<" ";

    return 0;
```

```
61 }
```

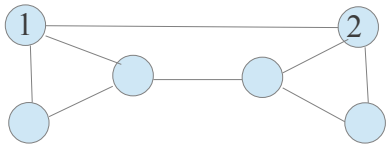
Решение на языке Паскаль

```
 1 var
 2     mm : array [0..99,0..199] of Boolean;
 3     mc : array [0..99,0..99] of Longint;
 4     mk : array [0..199] of Longint;
 5     rk, mn, rn, qn, res : array [0..99] of Longint;
 6     n, k, i, j, ck, it, t : Longint;
 7 begin
 8     assign(input, 'input.txt'); reset(input);
 9     assign(output, 'output.txt'); rewrite(output);
10
11     Read(n, k);
12
13     for i:=0 to k-1 do begin
14         mm[0, i] := true;
15         mk[i] := 1;
16     end;
17     mn[0] := k;
18     fillchar(res, sizeof(res), -1);
19     ck := (n-1) * k;
20     while ck > 0 do begin
21
22         fillchar(rk, sizeof(rk), -1);
23         for i:=0 to n-1 do
24             for j:=0 to k-1 do
25                 if (not mm[i, j]) and ((rk[i] < 0) or (mk[rk[i]] > mk[j])) then rk[i] := j;
26
27         fillchar(rn, sizeof(rn), -1);
28         for i:=0 to n-1 do begin
29             if rk[i] < 0 then continue ;
30             t := rk[i];
31             for j:=0 to n-1 do
32                 if (mm[j, t]) and ((rn[i] < 0) or (mn[rn[i]] > mn[j])) then rn[i] := j;
33         end;
34
35         fillchar(qn, sizeof(qn), -1);
36         for i:=0 to n-1 do begin
37             if rn[i] < 0 then continue ;
38             t := rn[i];
39             if (qn[t] < 0) or
                   (mc[t, qn[t]] < mc[t, i]) or
                   ((mc[t, qn[t]] = mc[t, i]) and (mn[qn[t]] > mn[i])) then qn[t] := i;
40         end;
41
42         for i:=0 to n-1 do begin
43             if qn[i] < 0 then continue ;
44             dec(ck);
45             inc(mc[qn[i], i]);
46             inc(mn[qn[i]]);
47             inc(mk[rk[qn[i]]]);
48             mm[qn[i], rk[qn[i]]] := true;
49             if mn[qn[i]] = k then res[qn[i]] := it+1;
50         end;
51         inc(it);
52     end;
53
54     for i:=1 to n-1 do Write(res[i],' ');
55
56 end.
```

Задача D. Отрезание ушей

Во время олимпиады не удалось решить данную задачу на 100 баллов. Зато оказалось, что жадное решение набирает 70 баллов (больше, чем любое решение других участников).
Суть жадного решения в следующем: берём любое ещё не удалённое ребро и продвигаемся от его концов в обе стороны пока степени вершин равны двум. Таким образом мы выбираем

какое-то ухо. Удаляем его. Одновременно надо ещё поддерживать степень вершин, чтобы вовремя остановиться и вывести «-1», когда у одной вершины степень станет равна 1. Такое решение «падает» на тесте:



Если первым ходом отрезать ухо $(1,2)$, то в графе появляется мост, и дальнейшие отрезания приведут к висячим вершинам.

Жадное решение на языке C++

```
#include <iostream>
#include <cstdio>
#include <memory.h>
#include <vector>
#include <set>

using namespace std;

const int V = 2e4+4;
const int E = 2e5+5;

int deg[V];
set<int> e[V];
bool del[V];
int cone=0, d=0;

void adde(int i, int j){
    e[i].insert(j);
    e[j].insert(i);
    ++deg[i];
    cone += (deg[i]==1);
    cone -= (deg[i]==2);
    ++deg[j];
    cone += (deg[j]==1);
    cone -= (deg[j]==2);
}

void dele(int i, int j){
    e[i].erase(j);
    e[j].erase(i);
    --deg[i];
    cone += (deg[i]==1);
    cone -= (deg[i]==0);
    --deg[j];
    cone += (deg[j]==1);
    cone -= (deg[j]==0);
}

vector<int> ans[E];

int main(){
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);

    int n,m,cntdel=0;
    scanf("%d%d",&n,&m);

    memset(deg, 0, sizeof(deg));

    for(int k=0;k<m;++k){
        int i,j;
        scanf("%d%d",&i,&j);
        adde(--i,--j);
    }

    vector<int> l, r;

```

```
    for(;;){
        if(cone){
            printf("-1\n");
            return 0;
        }
        int x = -1,y = -1;
        for(int i=0;i<n;++i) if(!del[i]){
            x = i;
            y = *e[i].begin();
            break;
        }

        if(x==-1 && y==-1) break;

        l.clear();
        r.clear();

        int px = x, py = y;

        while(e[y].size()==2){
            if(*e[y].begin()==px){
                px = y;
                y = *e[y].rbegin();
            }else{
                px = y;
                y = *e[y].begin();
            }
            r.push_back(px);
            if(y==x) break;
        }

        while(e[x].size()==2){
            if(*e[x].begin()==py){
                py = x;
                x = *e[x].rbegin();
            }else{
                py = x;
                x = *e[x].begin();
            }
            l.push_back(py);
            if(y==x) break;
        }

        ans[d].push_back(x);
        for(int k=(int)l.size()-1; k>=0; --k) ans[d].push_back(l[k]);
        for(int k=0; k<(int)r.size(); ++k) ans[d].push_back(r[k]);
        ans[d].push_back(y);

        for(int k=1;k<(int)ans[d].size();++k) dele(ans[d][k-1], ans[d][k]);
        for(int k=1;k<(int)ans[d].size()-1;++k){
            cntdel += !del[ans[d][k]];
            del[ans[d][k]] = true;
        }

        ++d;

        if(cntdel == n-1) break;
        if(cntdel==n){
            ans[d-1].resize(ans[d-1].size()/2+1);
            break;
        }
    }

    printf("%d\n",d);
    for(int i=0;i<d;++i){
        printf("%d",(int)ans[i].size()-1);
        for(int k=0;k<ans[i].size();++k) printf(" %d",ans[i][k]+1);
        printf("\n");
    }

    return 0;
}
```