**電話圈**

<cstdio> <vector> <string><map><cstring>

vector<string> names;

int ID(const string& s) {

for(int i = 0; i < names.size(); i++)

if(names[i] == s) return i;

names.push\_back(s);

return names.size() - 1;

}

const int maxn = 25 + 5;

int n, m, vis[maxn], d[maxn][maxn];

void dfs(int u) {

vis[u] = 1;

for(int v = 0; v < n; v++)

if(!vis[v] && d[u][v] && d[v][u]) {

printf(", %s", names[v].c\_str());

dfs(v);

}

}

int main() {

char s1[99], s2[99];

while(輸入n,m== 2 && n) {

//**n:人數 m:電話幾通**

names.clear();

memset(d, 0, sizeof(d));

for(int i = 0; i < n; i++) d[i][i] = 1;

while(m--) {

scanf("%s%s", s1, s2); //**s1s2人名**

d[ID(s1)][ID(s2)] = 1;

//**d:s1給s2打過電話** }

for(int k = 0; k < n; k++)

for(int i = 0; i < n; i++)

for(int j = 0; j < n; j++)

d[i][j] |= d[i][k] && d[k][j];

memset(vis, 0, sizeof(vis));

for(int i = 0; i < n; i++)

if(!vis[i]) {

printf("%s", names[i].c\_str());

dfs(i);

printf("\n");

}

while:} main:}

**噪音恐懼症**

***路徑上的 最大Weight 最小***

#include<cstdio><algorithm>

const int maxn = 100 + 5;

const int INF = 1000000000;

int d[maxn][maxn];

int main() {

int n, m, Q, u, v, w, kase = 0;

while(輸入n,m,Q == 3 && n) {

// 初始化

for(int i = 0; i < n; i++) {

d[i][i] = 0;

for(int j = i+1; j < n; j++) { d[i][j] = d[j][i] = INF; }

}

for(int i = 0; i < m; i++) {

scanf("%d%d%d", &u, &v, &w); u--; v--;

d[u][v] = min(d[u][v], w);

d[v][u] = d[u][v];

}

for(int k = 0; k < n; k++)

for(int i = 0; i < n; i++)

for(int j = 0; j < n; j++)

if(d[i][k] < INF && d[k][j] < INF)

d[i][j] = min(d[i][j], max(d[i][k], d[k][j]));

// 询问

while(Q--) {

scanf("%d%d", &u, &v); u--; v--;

if(d[u][v] == INF) printf("no path\n"); else printf("%d\n", d[u][v]);

}

}

}

***MST***

***F：find(int x)***

return x == parent[x] ? x : find(parent[x]);

***struct Point*** {double x, y;} p[755];

***struct Map*** { int a, b; double value;

} map[555555];

***F：cmp***

int main() {

for (i = 0; i < n; i ++)

for (j = i + 1; j < n; j ++) {

map[mapn].a = i;

map[mapn].b = j;

map[mapn ++].value = sqrt();

sort(map, map + mapn, cmp);

for (i = 0; i < mapn; i ++) {

int pa = find(map[i].a);

int pb = find(map[i].b);

if (pa != pb) {

parent[pa] = pb;

ans = map[i].value;

num ++;

if (num == n-s) break;

}

}

printf("%.2lf\n", ans);}}

***打飛機***

#include<algorithm>

<vector> <cstdio> <cstring> <cmath>

typedef long long ll;

typedef unsigned int uint;

typedef unsigned long long ull;

const double PI=acos(-1.0);

const double eps=1e-10;

int dcmp(double x)

{

if(fabs(x)<eps)return 0;

return x<0?-1:1;

}

struct Point

{

double x,y;

Point(){}

Point(double x,double y):x(x),y(y){}

Point operator+(const Point&p){return Point(x+p.x,y+p.y);}

Point operator-(const Point&p){return Point(x-p.x,y-p.y);}

Point operator\*(double p){return Point(x\*p,y\*p);}

Point operator/(double p){return Point(x/p,y/p);}

};

typedef Point Vector;

double Dot(Vector a,Vector b){return a.x\*b.x+a.y\*b.y;}

double Cross(Vector a,Vector b){return a.x\*b.y-a.y\*b.x;}

double Length(Vector a){return sqrt(Dot(a,a));}

double Angle(Vector a,Vector b){return acos(Dot(a,b)/Length(a)/Length(b));}

bool OnSegment(Point p,Point a,Point b)

{

if(dcmp(Cross(p-a,p-b)))return 0;

return dcmp(a.x-p.x)\*dcmp(b.x-p.x)<=0&&dcmp(a.y-p.y)\*dcmp(b.y-p.y)<=0;

}

typedef vector<Point> Polygon;

int isPointInPolygon(Point p,Polygon poly)

{

int wn=0;

int n=poly.size();

for(int i=0;i<n;i++)

{

if(OnSegment(p,poly[i],poly[(i+1)%n]))return 0;

int k=dcmp(Cross(poly[(i+1)%n]-poly[i],p-poly[i]));

int d1=dcmp(poly[i].y-p.y);

int d2=dcmp(poly[(i+1)%n].y-p.y);

if(k>0&&d1<=0&&d2>0)wn++;

if(k<0&&d2<=0&&d1>0)wn--;

}

return wn;

}

double v,b,g;

int n;

int main()

{

while(~scanf("%lf%lf%lf",&v,&b,&g))

{

if(v==0&&b==0&&g==0)break;

scanf("%d",&n);

Polygon p;

double my=0.0,x,y;

for(int i=0;i<n;i++)

{

scanf("%lf%lf",&x,&y);

my=max(my,y);

p.push\_back(Point(x,y));

}

int ok=0;

double T=dcmp(g)?2.0\*b/g:my/b;

for(double t=0.0;t<=T;t+=0.001)

{

Point tmp(-v\*t,b\*t-0.5\*g\*t\*t);

if(isPointInPolygon(tmp,p))

{

printf("%.2lf\n",t);

ok=1;

break;

}

}

if(!ok)puts("Miss!");

}

}

***類似二分***

int lo\_bound(int v){

int l=1,r=n,m;

while(l<r){

m = l+(r-l+1)/2;

if(a[m]>=v) r=m-1;

else l=m;

}

return l;

}

int up\_bound(int v){

int l=1,r=n,m;

while(l<r){

m = l+(r-l)/2;

if(a[m]<=v) l=m+1;

else r=m;

}

return r;

}

for(int i=0;i<q;i++){

if(b[i]<=a[1]) cout << "X ";

else cout << a[lo\_bound(b[i])] << " ";

if(b[i]>=a[n]) cout << "X" << endl;

else cout << a[up\_bound(b[i])] << endl;

}