

Ylgaýan adam (BOI1997)

Bir adam zyndandan gaçýar we polisiýa işgärlerinden sypmak maksady bilen kwadrat syrly otaga (labirint) girýär. Syrly otag ýörelýän ýollardan, gapylardan we diwarlardan düzülen. Her gapy iki ýa-da köp ýollary birleşdirýär. Otagyň bir girişi bar we ol şeýle hem otagyň çykyşy diýlip kabul edilýär. Ol ýer otagyň kenarynda ýerleşýär.

Polisiýa işgärleri, daşdan dolandyрма enjamyny ulanyp, girişden başga, otagyň içindeki islendik gapyny gulplap bilýärler. Başda, ähli gapylar açyk. Ylgaýan adamyň şu wagtky duran ýeri belli we ol ýer ýoluň ugrynda. Otagyň içinde diňe aşak, ýokaryk, saga we çeppe hereket edip bolýar. Ylgaýan adamyň diwaryň üstünde ýa-da gulply gapyda bolup bilmejegi bes-belli.

Sizden ol ylgaýan adamy otagyň içinde gapana düşürmek üçin polisiýa işgärleri haýyş edýärler. DİNE BIR GAPYNY gulplamaga hakyňyz bar we ol gapy giriş/çykyşdan başga ýer bolmaly. Otagyň bolup biljek iň uly meýdany 50x50.

GIRIŞ

Siziň programmaňyz girişi “INPUT.TXT” atly ASCII faýldan okamaly. Birinji setirde, bir pozitiw bitin san berlen, ýagny otagyň bir tarapyň ululygy. Ikinji setirde, iki sany bitin san berlen. Ol sanlar ylgaýan adamynyň duran ýerini aňladýar. Bitin sanlar biri-birinden boşluk bilen aýrylan. Birinji san adamyň duran ýeriniň setir nomeri, ikinji san bolsa sütün nomeri. Nomerleme, otagyň aşak-çep burjyndan başlaýar, ýagny ol ýer(1,1). Ondan soňky setirler (faýlyň soňuna çenli), otagyň içi barada maglumat saklaýar. D – simwoly gapyny, W – diwary, P – ýöräp bolýan ýoly we E – giriş/çykyşy aňladýar. Iki sany harpyň arasynda boşluk bar.

ÇYKYŞ

Siziň programmaňyz çykyşy “OUTPUT.TXT” faýlyna aşakdaky görnüşde etmeli: Birinji setirde “YES” sözi bolmaly, eger-de bir gapyny gulplap adamyny gapana düşürip bolýan bolsa. Egerde bolmasa onda “NO” diýip ýazdyrmaly. Egerde jogap YES bolsa, indiki setirde bir bitin san bolmaly. Ol san, şerti kanagatlandyryan gapylaryň sany. Ondan soňky setirlerde, şerti kanagatlandyryan gapylaryň ýerlerini bermeli. (hersi bir setirde).

MESELEM

INPUT.TXT	OUTPUT.TXT
5	YES
4 4	1
W W W W W	3 4
W W D P W	
W D P W W	
W P D P W	
W E W W W	

Her test üçin 3 sekunt berilýär.

Çözgüdi

Bu soragy çözmek üçin, “hemmesini-gör” metody ulanýarys, ýagny, ähli gapylary ýeke-ýekeden gulplap görmeli. Adama otagyň daşyna çykmak mümkin bolmasa, onda ol gapyny sanawa goşmaly.

```
Ähli i gapy üçin
{
    i gapyny gulpla.
    Adamynyň duran ýerinden çykyşa ýol tap.
    Eger ýol tapyp bolsa, Sanawa goş
    i gapyny aç.
}
```

Indiki mesele, gulplanan gapydan geçmän, çykyşa ýol tapmak. Bu meseläni birnäçe ýol bilen çözüp bolar. Ýöne biziň meselämiz üçin iň gowusy, DFS metodyny ulanmak. DFS-niň edýän işi, bir ýerden başlap, onuň goňşylaryna gitmek. Her gelen ýeriňe bellik goýmaly. Şol ýerik ikinji gezek gelmeli däl. Iň soňunda, adamynyň gelip bilýän hemme ýerlerine bellik goýulýar. Eger çykyşa bellik goýulan bolsa, onda ol adam i gapyny gulplanyňda hem otagdan gaçyp biljek eken.

```
Git(x, y)
{
    (x, y) bellik goý.

    Eger (x+1, y)-a (bellik goýulmadyk bolsa)
    we ((x+1, y) diwar ýa-da gulply gapy bolmasa)
        GIT(x+1, y);

    Eger (x-1, y)-a (bellik goýulmadyk bolsa)
    we ((x-1, y) diwar ýa-da gulply gapy bolmasa)
        GIT(x-1, y);

    Eger (x, y+1)-a (bellik goýulmadyk bolsa)
    we ((x, y+1) diwar ýa-da gulply gapy bolmasa)
        GIT(x, y+1);

    Eger (x, y-1)-a (bellik goýulmadyk bolsa)
    we ((x, y-1) diwar ýa-da gulply gapy bolmasa)
        GIT(x, y-1);
}
```

Indi bolsa bu çözüwimiziň işleýiş wagtyny hasaplaly. Bu ýerde näçe sany gapy bar bolsa, şonça gezek Git çagyrylar. Git $O(N^2)$ wagtda işleýär. Ýagny, otagyň her bir öýjüğine bir gezek barýar. Otagda $N*N$ sany öýjük bar. Gapy sany bolsa iň kän, $N*N$ bolup biler. (otagyň hemme ýeri gapy bolanda). Bu programmamyzyň wagtdaky ösüş tizligi $O(N^4)$. N iň uly 50 bolup bilýär. Programma 3 sekundyň içinde işini gutarar. Sebäbi N gaty kiçi we biziň ýazan çözüdimiz täsirli.

Ine programmamyz:

```
/*
    5th Balkan Olympiad in Informatics, 1997
    Day 1 Problem 1 The Running Man
    (C) Copyright, Atamyrat Hezretgulyev, 2000-2004.
*/

#include<stdio.h>
#include<string.h>

#define infile "runman.in"
#define outfile "runman.out"
#define MAXN 52

int M[MAXN][MAXN];
int n, rx, ry, nd=0, nans, ok;
int dx[MAXN*MAXN], dy[MAXN*MAXN];
int ax[MAXN*MAXN], ay[MAXN*MAXN];

void findpath(int x, int y)
{
    if(M[x][y]==2){ ok=1; return;}
    M[x][y]=5;

    if(M[x+1][y]%2==0 && !ok) findpath(x+1,y);
    if(M[x-1][y]%2==0 && !ok) findpath(x-1,y);
    if(M[x][y+1]%2==0 && !ok) findpath(x,y+1);
    if(M[x][y-1]%2==0 && !ok) findpath(x,y-1);
}

int main()
{
    int i,j;
    char ch,blank;

    FILE *fin=fopen(infile,"r");
    fscanf(fin,"%d%d%d",&n,&rx,&ry);
    fgetc(fin); //read newline char
    memset(M,1,sizeof(M));
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            {
                fscanf(fin,"%c%c",&ch,&blank);
                if(ch=='W') M[i][j]=1;
                if(ch=='P' || ch=='D') M[i][j]=0;
                if(ch=='E') M[i][j]=2;
                if(ch=='D')
                {
                    nd++;
                    dx[nd]=i;
                    dy[nd]=j;
                }
                if(!(ch=='W' || ch=='P' || ch=='E' || ch=='D'))
            {printf("Invalid Input!\n"); return 0;};
            }
    fclose(fin);

    for(i=1; i<=nd; i++)
    {
```

```

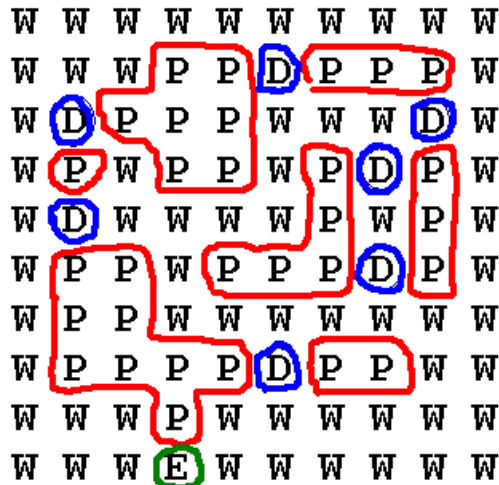
M[dx[i]][dy[i]]=1;
ok=0;
findpath(n-ry+1, rx);
for(int x=1; x<=n; x++)
    for(int y=1; y<=n; y++)
        if(M[x][y]==5) M[x][y]=0;
M[dx[i]][dy[i]]=0;
if(!ok)
{
    nans++;
    ax[nans]=dy[i];
    ay[nans]=n-dx[i]+1;
}
}

FILE *fout=fopen(outfile,"w");
if(nans)
{
    fprintf(fout,"YES\n%d\n",nans);
    for(i=1; i<=nans; i++) fprintf(fout,"%d %d\n",ax[i],ay[i]);
} else fprintf(fout,"NO\n");
fclose(fout);
return 0;
}

```

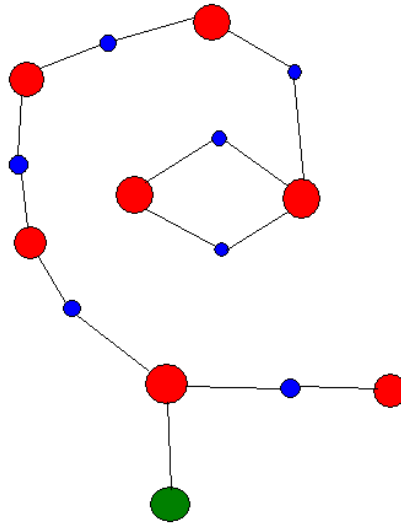
Cözgüt #2

Biz bu programmanyň işleýşini has hem çaltlandyryp bileris.



Şekil 1

Egerde şekil 1-daky ýaly otag berilen bolsa, onda ol otagy grafa öwürip bileris. Çykyş we her gapylar grafyň bir depesi, P toplulyklar bolsa, her toplum bir depe. Olaryň arasynda birleşme bolýar, egerde olar goňşy bolsa. Ýokarky otag üçin onuň grafy şekil 2-däki ýaly bolar. Soňy hem çözgüt 1-daky ýaly. Her gapylary gulplap görýäs. Ýöne çykyşa baryp bolýandygyny barlamak has tiz. Bir ädimde bütün toplumy geçip bolýar. Wagtdaky ösüş tizligi meňzeş, $O(N^4)$, ýöne, beýleki çözüde garanynda köp hili otagnar üçin jogaby has çalt tapýar.



Şekil 2

İne programmamyz:

```

/*
    5th Balkan Olympiad in Informatics, 1997
    The Running Man (Solution #2)
    (C) Copyright, Atamyrat Hezretgulyev, 2000-2004.
*/

#include<stdio.h>
#include<string.h>

#define infile "runman.in"
#define outfile "runman.out"
#define maxn 52
#define maxc maxn*maxn

struct TCell
{
    int id;
    int type;    // 0-wall 1-path 2-door 3-exit
} M[maxn][maxn];

int alist[maxc][maxc]; // adjacency list
int nd=0,np=0; // number of doors, number of paths
int used[maxc];
int n,rx,ry,ex,ey, start;
int nans=0, ax[maxc], ay[maxc],dx[maxc], dy[maxc];

void readinput ()
{
    int i,j;
    char c,blank;

    FILE *fin=fopen(infile,"r");
    fscanf(fin,"%d%d%d%c",&n,&rx,&ry,&blank);
    memset(M,0,sizeof(M));
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
        {
            fscanf(fin,"%c%c",&c,&blank);
            if(c=='W') M[i][j].type=0;
            if(c=='P') M[i][j].type=1;
            if(c=='D')

```

```

        {
            nd++;
            M[i][j].type=2;
            M[i][j].id=nd+1;
            dx[nd]=j;
            dy[nd]=n-i+1;
        }
        if(c=='E')
        {
            M[i][j].type=3;
            M[i][j].id=1;
            ex=i; ey=j;
        }
    }
    fclose(fin);
}

void label(int x, int y)
{
    M[x][y].id=nd+1+np;
    if(M[x+1][y].type==1 && M[x+1][y].id==0) label(x+1,y);
    if(M[x-1][y].type==1 && M[x-1][y].id==0) label(x-1,y);
    if(M[x][y+1].type==1 && M[x][y+1].id==0) label(x,y+1);
    if(M[x][y-1].type==1 && M[x][y-1].id==0) label(x,y-1);
}

void findcomp()
{
    int i,j;
    np=0;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            if(M[i][j].type==1 && M[i][j].id==0)
            {
                np++;
                label(i,j);
            }
    start=M[n-ry+1][rx].id;
    // output (debug)
    /*
    printf("\n\n\n");
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)    printf("%4d",M[i][j].id);
        printf("\n");
    }
    */
}

void buildgraph()
{
    int i,j,u,v;

    memset(alist,0,sizeof(alist));
    for(i=1; i<n; i++)
        for(j=1; j<n; j++)
        {
            if(M[i][j].id!=0 && M[i+1][j].id!=0)
            {
                u=M[i][j].id;
                v=M[i+1][j].id;
                if(u!=v)
                {

```

```

        // add edge (u,v) and (v,u)
        alist[u][0]++;
        alist[u][alist[u][0]]=v;
        alist[v][0]++;
        alist[v][alist[v][0]]=u;
    }
}
if(M[i][j].id!=0 && M[i][j+1].id!=0)
{
    u=M[i][j].id;
    v=M[i][j+1].id;
    if(u==v) continue;
    // add edge (u,v) and (v,u)
    alist[u][0]++;
    alist[u][alist[u][0]]=v;
    alist[v][0]++;
    alist[v][alist[v][0]]=u;
}
}
}
/*
for(i=1; i<=1+nd+np; i++)
{
    printf("%3d (%3d) ",i,alist[i][0]);
    for(j=1; j<=alist[i][0]; j++) printf(" --> %3d",alist[i][j]);
    printf("\n");
}
*/
}

int reachable()
{
    int queue[maxn*maxn]={0,start};
    int head=1, tail=0;
    int u,v;

    used[start]=1;
    while(tail<=head)
    {
        tail++;
        u=queue[tail];
        for(v=1; v<=alist[u][0]; v++)
            if(used[alist[u][v]]==0)
            {
                head++;
                used[alist[u][v]]=1;
                queue[head]=alist[u][v];
                if(alist[u][v]==1) return 1;
            }
    }
    return 0;
}

void search()
{
    int i;

    nans=0;
    for(i=2; i<=nd+1; i++) // try locking all doors
    {
        memset(used,0,sizeof(used));
        used[i]=1;
        if(!reachable())
        {

```

```

        nans++;
        ax[nans]=dx[i-1];
        ay[nans]=dy[i-1];
    }
}

void writeoutput ()
{
    int i;
    FILE *fout=fopen(outfile,"w");
    if(nans)
    {
        fprintf(fout,"YES\n%d\n",nans);
        for(i=1; i<=nans; i++) fprintf(fout,"%d %d\n",ax[i],ay[i]);
    } else fprintf(fout,"NO\n");
    fclose(fout);
}

int main()
{
    readinput();
    findcomp();
    buildgraph();
    search();
    writeoutput();
    return 0;
}

```

Atamyrat Hezretgulyev
atamyrat_hezretgulyev@yahoo.com