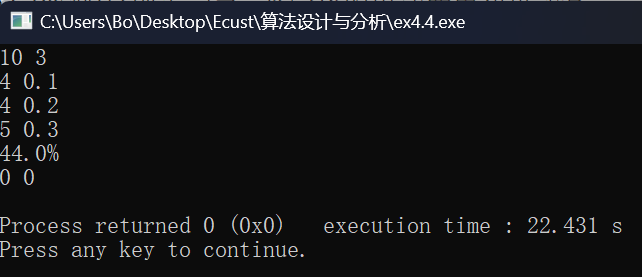
4-4



#include<iostream>

#include<algorithm>

#include<cstring>

using namespace std;

const int maxn=1e4+9;

int n,m,a[maxn];

double b[maxn],dp[maxn]; //dp[i]：预算为i，得到一份offer的最大概率

int main()

{

while(cin>>n>>m) //预算为n元 m个学校

{

if(n+m==0)

break;

memset(dp,0,sizeof(dp));

for(int i=1; i<=m; i++)

cin>>a[i]>>b[i]; //申请第i个学校的费用 和通过的概率

for(int i=1; i<=m; i++)

{

for(int j=n; j>=a[i]; j--) //类似01背包

{

dp[j]=max(dp[j],1-(1-dp[j-a[i]])\*(1-b[i]));

}

}

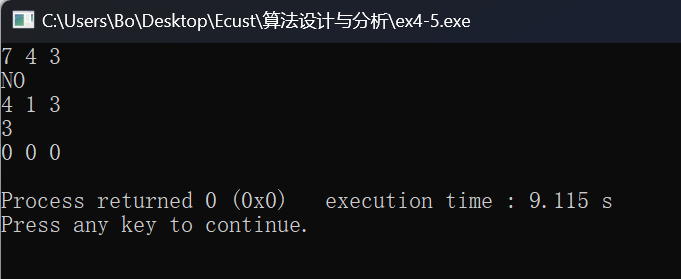
printf("%.1f%%\n",dp[n]\*100);

}

return 0;

}

4-5



#include<cstdio>

#include<iostream>

using namespace std;

int gcd(int a,int b)

{

return b?gcd(b,a%b):a;

}

int main()

{

int a,b,c;

while(scanf("%d%d%d",&a,&b,&c),a+b+c)

{

a/=gcd(b,c);

if(a&1)printf("NO\n");

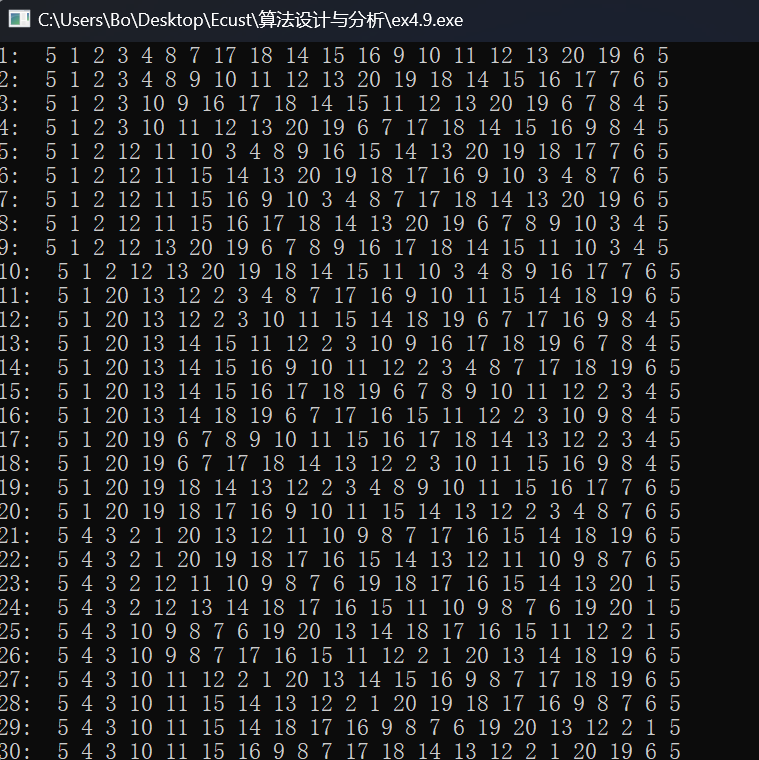
else printf("%d\n",a-1);

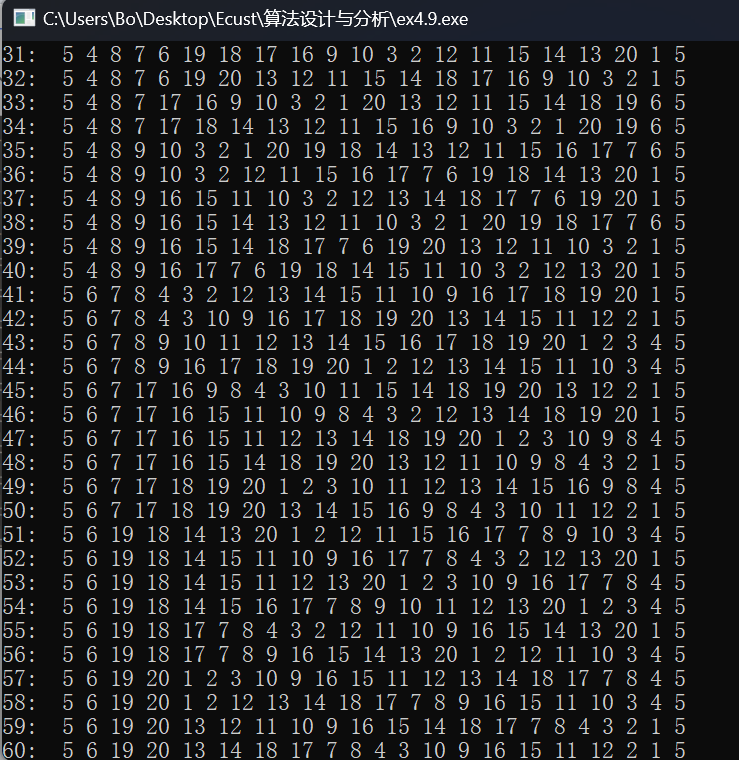
}

return 0;

}

4-9





#include<iostream>

#include<cstdio>

#include<cstring>

using namespace std;

int a[25][3];

bool b[25];

int c[25];

int k;

void dfs(int m, int step)

{

c[step] = m;

if(step == 19 )

{

int ans = 0;

for(int j = 0; j < 3; j++)

{

if(a[m][j] == c[0])

ans = 1;

}

if(ans == 0)

return ;

printf("%d: ", k);

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

{

printf("%d ", c[i]);

}

printf("%d\n", c[0]);

k++;

return;

}

for(int i = 0; i < 3; i++)

{

if(b[a[m][i]] == 0)

{

b[a[m][i]] = 1;

dfs(a[m][i], step+1);

b[a[m][i]] = 0;

}

}

}

int main()

{

int i, j, m;

for(i = 1; i <= 20; i++)

{

for(j = 0; j < 3; j++)

{

scanf("%d", &a[i][j]);

}

}

while(scanf("%d", &m)!=EOF){

if(m == 0)

break;

k = 1;

memset(b, 0, sizeof(b));

b[m] = 1;

dfs(m, 0);

}

return 0;

}

4-7



#include<stdio.h>

#include<queue>

#include<string.h>

#include<algorithm>

using namespace std;

int book[2][15][15];

char a[2][15][15];

struct node

{

int x,y,z,step;

};

int main()

{

int i,j,k,f,n,m,t,T;

int ta,tb,tc,tx,ty,tz;

int next [4][2]= {0,1,1,0,0,-1,-1,0};

scanf("%d",&T);

while(T--)

{

scanf("%d%d%d",&n,&m,&t);

getchar();

for(i=0; i<2; i++)

{

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

{

for(k=0; k<m; k++)

scanf("%c",&a[i][j][k]);

getchar();

}

if(i!=1) getchar();//这个吸收中间那个空白行，注意只有一个空白行。

}

for(i=0; i<2; i++)

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

{

for(k=0; k<m; k++)

if(a[i][j][k]=='P')//记录公主的位置，顺便把'P'改为'.',(纯属个人习惯)。

{

ta=i;

tb=j;

tc=k;

a[i][j][k]='.';

}

else if(a[i][j][k]=='#'&&a[i^1][j][k]=='#')//如果这一层是#，另一层还是#，直接标记为墙就行。

{

a[0][j][k]='\*';

a[1][j][k]='\*';

}

else if(a[i][j][k]=='#'&&a[i^1][j][k]=='\*')//这一层是#，另一层是墙，这样会撞死（骑士也不傻）。

{

a[0][j][k]='\*';

a[1][j][k]='\*';

}

}

memset(book,0,sizeof(book));

book[0][0][0]=1;

queue<node>Q;

node p,q;

f=0;

p.x=0,p.y=0,p.z=0,p.step=0;

Q.push(p);

while(!Q.empty())

{

p=Q.front();

Q.pop();

if(p.x==ta&&p.y==tb&&p.z==tc&&p.step<=t&&f==0)

{

f=1;

// printf("%d\n",p.step);

printf("YES\n");

break;

}

for(i=0; i<4; i++)

{

tx=p.x;

ty=p.y+next[i][0];

tz=p.z+next[i][1];

if(a[tx][ty][tz]=='\*'||book[tx][ty][tz]==1||tx<0||tx>1||ty<0||ty>n-1||tz<0||tz>m-1)

continue;

if(a[tx][ty][tz]=='.')

{

book[tx][ty][tz]=1;

q.x=tx,q.y=ty,q.z=tz;

q.step=p.step+1;

Q.push(q);

}

else if(a[tx][ty][tz]=='#')

{

book[tx][ty][tz]=1;

if(tx==0)//传送门可以换层（(⊙o⊙)哦，传送门是什么东东，怎么听着这么熟悉）。

tx+=1;

else tx-=1;

if(a[tx][ty][tz]=='\*'||book[tx][ty][tz]==1||tx<0||tx>1||ty<0||ty>n-1||tz<0||tz>m-1)

continue;

book[tx][ty][tz]=1;

q.x=tx,q.y=ty,q.z=tz;

q.step=p.step+1;

Q.push(q);

}

}

}

while(!Q.empty())

Q.pop();

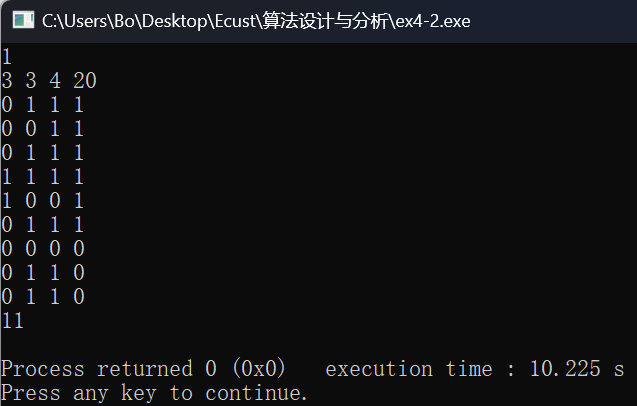
if(!f) printf("NO\n");

}

return 0;

}

4-2



#include<stdio.h>

#include<string.h>

#include<queue>

using namespace std;

int a,b,c,t;

int map[51][51][51],dis[6][3]={{1,0,0},{-1,0,0},{0,1,0},{0,-1,0},{0,0,1},{0,0,-1}};

bool biao[51][51][51];

struct ch

{

int x,y,z,step;

};

bool panduan(int x,int y,int z)

{

if(x<0||y<0||z<0||x>=a||y>=b||z>=c)return 0;

if(biao[x][y][z]||map[x][y][z])return 0;

return 1;

}

int bfs(int x,int y,int z)

{

queue<ch>ha;

ch sta,next;

int i,j;

sta.x=x;

sta.y=y;

sta.z=z;

sta.step=0;

biao[x][y][z]=1;

ha.push(sta);

while(!ha.empty())

{

sta=ha.front();

ha.pop();

if(sta.x==a-1&&sta.y==b-1&&sta.z==c-1)

{

return sta.step;

}

if(sta.step>t)

{

return -1;

}

for(i=0;i<6;i++)

{

next.x=sta.x+dis[i][0];

next.y=sta.y+dis[i][1];

next.z=sta.z+dis[i][2];

if(panduan(next.x,next.y,next.z))

{

next.step=sta.step+1;

biao[next.x][next.y][next.z]=1;

ha.push(next);

}

}

}

return -1;

}

int main()

{

int i,j,z,n;

scanf("%d",&n);

while(n--)

{

scanf("%d%d%d%d",&a,&b,&c,&t);

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

{

for(j=0;j<b;j++)

{

for(z=0;z<c;z++)

{

scanf("%d",&map[i][j][z]);

}

}

}

memset(biao,false,sizeof(biao));

printf("%d\n",bfs(0,0,0));

}

return 0;

}