



高斯消元：

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 1005

#define eps 1e-6

using namespace std;

ll n,m;

double a[mn][mn],ans[mn];

ll gauss(ll n,ll m)

{ll c=0,r,i,j;

for(r=0;r<n&&c<m;r++,c++)

{

ll maxr=r;

for(i=r+1;i<n;i++)

if(fabs(a[i][c])>fabs(a[maxr][c]))maxr=i;

if(maxr!=r)swap(a[r],a[maxr]);

if(fabs(a[r][c])<eps){r--;continue;}

for(i=r+1;i<n;i++)

if(fabs(a[i][c])>eps)

{

double k=a[i][c]/a[r][c];

for(j=c;j<m+1;j++)a[i][j]-=a[r][j]\*k;

a[i][c]=0;

}

}

for(i=r;i<m;i++)if(fabs(a[i][c])>eps)return -1;//无解

if(r<m)return m-r;//返回自由元个数

for(i=m-1;i>=0;i--)

{

for(j=i+1;j<m;j++)a[i][m]-=a[i][j]\*ans[j];

ans[i]=a[i][m]/a[i][i];

}

return 0;//有唯一解

}

int main()

{

ll x,y,z,i,j,k;

char ch;

cin>>n;

for(i=0;i<n;i++)//这个板子的数组是从0开始的

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

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

ll bj=gauss(n,n);

if(bj!=0){cout<<"No Solution";return 0;}

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

printf("%.2lf\n",ans[i]);

return 0;

}

高速约旦消元：

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 105

using namespace std;

ll ans,n,m;

double a[mn][mn];

int main()

{

ll x,y,z,i,j,k;

char ch;

cin>>n;

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

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

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

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

{

ll maxx=i;

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

if(fabs(a[j][i])>fabs(a[maxx][i]))

maxx=j;

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

swap(a[i][j],a[maxx][j]);

if(a[i][i]==0){printf("No Solution");return 0;}

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

if(j!=i)

{

double tmp=a[j][i]/a[i][i];

for(k=i+1;k<=n+1;k++)

a[j][k]-=a[i][k]\*tmp;

}

}

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

printf("%.2lf\n",a[i][n+1]/a[i][i]);

return 0;

}

下面这个代码是异或方程组，因为可以用bitset进行优化，特地放在这里作为参考

bitset<mn>x[mn],y[mn],a[mn];

void gauss(ll n,ll m)

{ll c=0,r,i,j;

for(r=0;r<n&&c<m;r++,c++)

{

ll maxr=r;

for(i=r+1;i<n;i++)

if(a[i][c]>a[maxr][c])maxr=i;

if(maxr!=r)swap(a[r],a[maxr]);

if(a[r][c]==0){r--;continue;}

for(i=r+1;i<n;i++)

if(a[i][c]>0)

{

bitset<mn>k=a[i][c]^a[r][c];

a[i]=a[i]^a[r]^k;//bitset优化

// for(j=c;j<m+1;j++)a[i][j]=a[i][j]^a[r][j]^k;

a[i][c]=0;

}

}

if(r<=m)ans+=m-r;//这个题ans是求解自由元个数的

}