矩形树定理

算法简介

一种生成树的计算定理，时间复杂度 $O\left(n^3\right)$

算法实现

无向图

定义生成树的权值为所有该生成树中所有边权的乘积，则有如下结论：

邻接矩阵 D 中 $d_{i,i}=$ 所有与节点 i 相连的边的权值和$d_{i,j}=0(i\neq j)$

邻接矩阵 L 中 $d_{i,j}=edge[i][j].w$ (注意无向图中 $d_{i,j}=d_{j,i}$)

记基尔霍夫矩阵 $K=D-L$$K'$ 为 K 去掉第 i 行与第 i 列得到的余子式(i 可以任取)。

则有 $det(K')=$ 所有生成树的权值和。特别地，当所有边权为 1 时所有生成树的权值和等于生成树个数。

有向图

邻接矩阵 L 定义不变(但要注意边的有向性)。

如果邻接矩阵 D 中 $d_{i,i}=$ 节点 i 的所有入边的权值和。

记 K' 为K 去掉第 i 行与第 i 列得到的余子式，则 $det(K')=$ 所有以节点 i 为根的外向树(边从根指向叶子节点)的权值和。

如果邻接矩阵 D 中 $d_{i,i}=$ 节点 i 的所有出边的权值和。

记 K' 为K 去掉第 i 行与第 i 列得到的余子式，则 $det(K')=$ 所有以节点 i 为根的内向树(边从叶子节点指向根)的权值和。

代码模板

洛谷p6178

给定一个 n 个节点 m 条带权边的图，输入 t 表示图是否为有向图。

求其所有不同生成树的权值之和(如果是有向图，则求以 1 为根的外向树)，对 10^9+7 取模。

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
```

```
#include <algorithm>
#include <string>
#include <sstream>
#include <cstring>
#include <cctype>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <stack>
#include <queue>
#include <ctime>
#include <cassert>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset(a,b,sizeof(a))
using namespace std;
typedef long long LL;
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline char get_char(){
    char c=getchar();
    while(c==' '||c=='\n'||c=='\r')c=getchar();
    return c;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAX_size=305,mod=1e9+7;
struct Matrix{
    int ele[MAX_size][MAX_size];
};
int Inv(int x,int p){
    int ans=1,base=x,k=p-2;
```

```
    while(k){
        if(k&1)
        ans=1LL*ans*base%p;
        base=1LL*base*base%p;
        k>>=1;
    }
    return ans;
}
int det(Matrix a,int n,int mod){
    int ans=1;
    _rep(i,2,n){
        int pos=i;
        _rep(j,i,n)if(a.ele[j][i]){pos=j;break;}
        if(!a.ele[pos][i])return 0;
        if(pos!=i){_rep(j,i,n) swap(a.ele[i][j],a.ele[pos][j]);ans=mod-
ans;}
        ans=1LL*ans*a.ele[i][i]%mod;
        int k=Inv(a.ele[i][i],mod);
        _rep(j,i,n)a.ele[i][j]=1LL*a.ele[i][j]*k%mod;
        _rep(j,i+1,n)for(int k=n;k>=i;k--)
        a.ele[j][k]=(a.ele[j][k]-1LL*a.ele[j][i]*a.ele[i][k])%mod;
    }
    return (ans+mod)%mod;
}
int main()
{
    int n=read_int(),m=read_int(),t=read_int(),u,v,w;
    Matrix x;
    mem(x.ele,0);
    if(t==0){
        while(m--){
            u=read_int(),v=read_int(),w=read_int();
            if(u==v)continue;
x.ele[u][u]=(x.ele[u][u]+w)%mod,x.ele[v][v]=(x.ele[v][v]+w)%mod;
            x.ele[u][v]=(x.ele[u][v]-w)%mod,x.ele[v][u]=(x.ele[v][u]-
w)%mod;
        }
    }
    else{
        while(m--){
            u=read_int(),v=read_int(),w=read_int();
            if(u==v)continue;
            x.ele[u][v]=(x.ele[u][v]-
w)%mod,x.ele[v][v]=(x.ele[v][v]+w)%mod;
        }
    }
    enter(det(x,n,mod));
    return 0;
}
```

算法练习

习题一

洛谷p3317

题意

给定一个 n 个点的完全图，表示 n 个城市，该地区经过了一场洪水，城市之间的道路受损。

输入一个 $n\times n$ 矩阵，矩阵元素 $p_{i,j}$ 表示城市 i,j 之间道路依然连通的概率。

问经过洪水后该地区所有道路恰好构成一棵树的概率。

输入保证 $p_{i,j}=p_{j,i},p_{i,i}=0$

题解

若将 $p_{i,j}$ 作为边 i,j 的权值套用矩阵树定理，设 E 为总边集 T 为生成树边集，则有

$$P=\sum_T\prod_{e\in E-T}(1-p_e)\prod_{e\in T}p_e=\prod_{e\in E}(1-p_e)\sum_T\prod_{e\in T}\frac {p_e}{1-p_e}$$

最后关于 $p_e=1$ 的情况，可以考虑缩点，或者令 $p_e=1-\varepsilon$

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <algorithm>
#include <string>
#include <sstream>
#include <cstring>
#include <cctype>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <stack>
#include <queue>
#include <ctime>
#include <cassert>
#define _for(i,a,b) for(int i=(a);i<(b);++i)
#define _rep(i,a,b) for(int i=(a);i<=(b);++i)
#define mem(a,b) memset(a,b,sizeof(a))
using namespace std;
typedef long long LL;
```

```
inline int read_int(){
    int t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline LL read_LL(){
    LL t=0;bool sign=false;char c=getchar();
    while(!isdigit(c)){sign|=c=='-';c=getchar();}
    while(isdigit(c)){t=(t<<1)+(t<<3)+(c&15);c=getchar();}
    return sign?-t:t;
}
inline char get_char(){
    char c=getchar();
    while(c==' '||c=='\n'||c=='\r')c=getchar();
    return c;
}
inline void write(LL x){
    register char c[21],len=0;
    if(!x)return putchar('0'),void();
    if(x<0)x=-x,putchar('-');
    while(x)c[++len]=x%10,x/=10;
    while(len)putchar(c[len--]+48);
}
inline void space(LL x){write(x),putchar(' ');}
inline void enter(LL x){write(x),putchar('\n');}
const int MAX_size=55;
const double eps=1e-8;
struct Matrix{
    double ele[MAX_size][MAX_size];
};
double det(Matrix a,int n){
    double ans=1.0;
    _rep(i,2,n){
        int pos=i;
        _rep(j,i+1,n)if(fabs(a.ele[j][i])>fabs(a.ele[pos][i]))pos=j;
        if(fabs(a.ele[pos][i])<eps)return 0.0;
        if(pos!=i){_rep(j,i,n)swap(a.ele[i][j],a.ele[pos][j]);ans=-ans;}
        ans*=a.ele[i][i];
        for(int j=n;j>=i;j--)a.ele[i][j]/=a.ele[i][i];
        _rep(j,i+1,n)for(int k=n;k>=i;k--)
        a.ele[j][k]=a.ele[j][k]-a.ele[j][i]*a.ele[i][k];
    }
    return ans;
}
double a[MAX_size][MAX_size];
int main()
{
    int n=read_int();double ans=1.0;
    Matrix x;
    mem(x.ele,0);
```

```
    _rep(i,1,n)
    _rep(j,1,n){
        scanf("%lf",&a[i][j]);
        if(fabs(1.0-a[i][j])<eps)
        a[i][j]=1.0-eps;
        if(i>j)
        ans*=1.0-a[i][j];
        a[i][j]/=1.0-a[i][j];
    }
    _rep(i,1,n)
    _rep(j,1,n)
    x.ele[i][j]-=a[i][j],x.ele[j][j]+=a[i][j];
    printf("%lf",ans*det(x,n));
    return 0;
}
```

习题二

洛谷p4208

题意

现在给出了一个简单无向加权图，求这个图中有多少个不同的最小生成树，结果对 31011 的模。

题解

先给出最小生成树的两个性质

1. 如果 A,B 都是 G 的最小生成树，则将 A,B 中所有边权从小到大排序，将得到相同结果。
2. 如果 A,B 都是 G 的最小生成树，则删去 A,B 中权值超过 w 的边后，A,B 图连通性完全相同。(w 取值任意)

性质一证明：