

第一场周赛T4

类似爬经典楼梯这道题,我们容易写出方程 $f[i] = (f[i - k] + f[i - k + 1] + \dots + f[i - 1]) \bmod 20211114$
用矩阵快速优化一下, 难点在于构造矩阵

$$f_n = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} * \begin{bmatrix} f_{n-1} \\ f_{n-2} \\ f_{n-3} \\ \vdots \\ f_{n-k+1} \\ f_{n-k} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}^{n-k} * \begin{bmatrix} f_k \\ f_{k-1} \\ f_{k-2} \\ \vdots \\ f_2 \\ f_1 \end{bmatrix}$$

代码如下

```
#include<bits/stdc++.h>

using namespace std;

const int N=55;
#define ll long long
const int mod=20211114;
int m;
struct Matrix{
    ll a[N][N];
    Matrix(){
        memset(a,0,sizeof(a));
    }
    void init(){
        for(int i=1;i<=m;i++) a[i][i]=1;
    }
    Matrix operator * (const Matrix &A){
        Matrix B;
        for(int i=1;i<=m;i++)
            for(int j=1;j<=m;j++)
                for(int k=1;k<=m;k++){
                    B.a[i][j]=(B.a[i][j]+a[i][k]*A.a[k][j])%mod;
                }
        return B;
    }
};

Matrix quick(Matrix A,ll b){
    Matrix res;
    res.init();
    for(;b;b>>=1){
        if(b&1) res=res*A;
        A=A*A;
    }
    return res;
}
```

```

}

ll n;
int f[N];
int main(){
    cin>>n>>m;
    f[0]=1;
    for(int i=1;i<=m;i++)
        for(int j=1;j<=min(i,m);j++)
            f[i]=(f[i]+f[i-j])%mod;
    if(n<=m){
        cout<<f[n]<<endl;
    }
    else{
        Matrix A,F;
        for(int i=1;i<=m;i++) A.a[1][i]=1;
        for(int i=2;i<=m;i++)
            A.a[i][i-1]=1;
        for(int i=1;i<=m;i++) F.a[i][1]=f[m-i+1];
        F=quick(A,n-m)*F;
        cout<<F.a[1][1]<<endl;
    }
    return 0;
}

```

[第一场周赛 T5](#)

前面50%用状态压缩动态规划，后50%用模拟退火等近似算法解决TSP问题

```

#include<bits/stdc++.h>

using namespace std;

int dis[50][50];
int n;
int f[1<<16][16];
int ans,last;
int w[50];

const double eps=1e-15,delta=0.985;
int fun(){
    int res=0;
    for(int i=0;i<n-1;i++)
        res+=dis[w[i]][w[i+1]];
    return res;
}
//后50%用模拟退火
void sa(){
    random_shuffle(w+1,w+n-1);
    for(double t=1000;t>eps;t*=delta){
        int last=fun();
        int x=rand()%(n-2)+1,y=rand()%(n-2)+1;
        swap(w[x],w[y]);
        int now=fun();
        int de=now-last;
        if(de<0){
            last=now;
        }
    }
}

```

```

        ans=min(ans,now);
    }
    else if(exp(-1.0*de/t)/RAND_MAX>rand()) //以一定的概率接受劣解
        last=now;
    else
        swap(w[x],w[y]);
}
}

void solve2(){
    for(int i=0;i<n;i++) w[i]=i;
    ans=last=fun();
    for(int i=0;i<50;i++) sa();
    cout<<ans<<endl;
}

//前50% 用动态规划
void solve1(){
    memset(f,0x3f,sizeof(f));
    f[1][0]=0;
    for(int i=0;i<(1<<n);i++){
        for(int j=0;j<n;j++) if(i>>j&1)
            for(int k=0;k<n;k++) if((i^(1<<j))>>k&1)
                f[i][j]=min(f[i][j],f[(i^(1<<j))][k]+dis[k][j]);
    }
    cout<<f[(1<<n)-1][n-1]<<endl;
}

int main(){
    cin>>n;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++) cin>>dis[i][j];
    if(n<=15) solve1();
    else solve2();
    return 0;
}

```

参考解TSP问题参考资料

[模拟退火学习笔记](#)

[禁忌搜索学习笔记](#)