

# P4017 最大食物链计数(拓扑排序+简单dp)

## [传送门](#)

思路：拓扑排序+简单 $dp$ 。显然每个最大食物链起点为是入度为0的点，终点是出度为0的点，这样我们只需要统计每个出度为0的点的最大食物链贡献，最后求和。

设 $ans[i]$ 为从所有入度为0点到达该点的路径数。

显然对于相连边有转移方程： $ans[v] += ans[u]$

时间复杂度： $O(n + m)$

**AC代码：**

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

using namespace std;

typedef long long ll;

const int N=5e3+5,mod=80112002;

#define mst(a) memset(a,0,sizeof a)

int in[N],n,m,out[N];

ll ans[N];

vector<int>e[N];

ll toposort() {

    queue<int>q;

    for(int i=1;i<=n;i++) if(!in[i]) q.push(i),ans[i]=1;

    while(q.size()) { //拓扑排序板子

        int u=q.front();q.pop();

        for(auto v:e[u]) {

            ans[v]=(ans[v]+ans[u])%mod;//状态转移

            in[v]--;

            if(!in[v]) q.push(v);

        }

    }

    ll res=0;

    for(int i=1;i<=n;i++)//求和

        if(!out[i]) res=(res+ans[i])%mod;

    return res;

}

int main() {
```

```
scanf("%d%d",&n,&m);  
for(int i=1,u,v;i<=m;i++){  
    scanf("%d%d",&u,&v);  
    e[u].push_back(v),e[v].push_back(u);  
    out[u]++,in[v]++;  
}  
printf("%lld\n",toposort());  
return 0;  
}
```