[CF1376B]Maximal Independent Set

Algorithm	NP完全问题 贪心
Created	@Jun 25, 2020 9:23 PM
Difficulty	省选/NOI-
Related to 近期更新 (Property)	
URL	https://codeforces.com/contest/1376/problem/B1

题目链接：

Problem - B1 - Codeforces

You are given an undirected graph $$$G=(V,E)$$$. Your task is to find such a maximal subset of vertices that no two vertices in the subset are connected with an edge from

https://codeforces.com/contest/1376/problem/B1

CODEFORCES β Sponsored by Telegram

https://codeforces.ml/contest/1376/problem/B1 （镜像站）

题解：

从某种程度上来说，本题一直没有快速解（如果你想挑战一下图灵奖，可以点一下这里）

那么步入正题

仔细分析一下本问题，可发现以下的贪心策略（此贪心策略有问题，反例大家自己举）：

每次删掉度最大的结点以及与其相关的边，直到图中所有的结点度都为0，此时，剩余的结点就是该树的最大独立集。

然后，放手吧，

让机器运行个几小时也差不多了……

附上代码：

```
#include <bits/stdc++.h>
using namespace std;
int n,m,nn;
struct edge
{
    int a,b;
}ed[5000000];
struct node
{
    int c,index;
}no[500000];
void deletee(int x)  //删除一条边
{
    for(int i=x;i<m;i++)
        ed[i]=ed[i+1];
    m--;
}
void deleten(int x)  //删除一个节点
{
    for(int i=1;i<=m;i++)
        if(ed[i].a==x||ed[i].b==x)
            deletee(i);  //删除所有与该节点相关的边
    for(int i=x;i<n;i++)
        no[i]=no[i+1];
    n--;
}
bool all()  //检查是否每个节点的度都为零
{
    for(int i=1;i<=n;i++)
        if(no[i].c!=0)
            return 0;
    return 1;
}
set<int> is;  //最大独立子集
```

```
int main()
{
    freopen("b2.in","r",stdin);
    freopen("b2.out","w",stdout);
    int a,b;
    scanf("%d%d",&n,&m);
    nn=n;
    for(int i=1;i<=n;i++)
        no[i].index=i;
    for(int i=1;i<=m;i++)
    {
        scanf("%d%d",&a,&b);
        ed[i].a=a;ed[i].b=b;  //插入边
        no[a].c++;no[b].c++;  //边两边的度分别加一
    }
    while(!all())  //一直删除最大度节点，直到所有节点的度都为零
    {
        int maxn=0,man=0;
        for(int i=1;i<=n;i++)  //找出最大度
            if(no[i].c>maxn) {
                maxn = no[i].c;
                man = i;
            }
        deleten(man);
    }
    printf("%d\n",n);
    for(int i=1;i<=n;i++)
        is.insert(no[i].index);
    for(int i=1;i<nn;i++)
    {
        if(is.find(i)==is.end())
            printf("0 ");
        else
            printf("1 ");
    }
    if(is.find(nn)==is.end())
        printf("0\n");
    else
        printf("1\n");
    return 0;
}
```

那么，本题有没有正确的解法呢？有是肯定有，但是本蒟蒻不会。会的可以补充或发到邮箱lvherui.lx@gmail.com 谢谢路过大佬的指点 。