

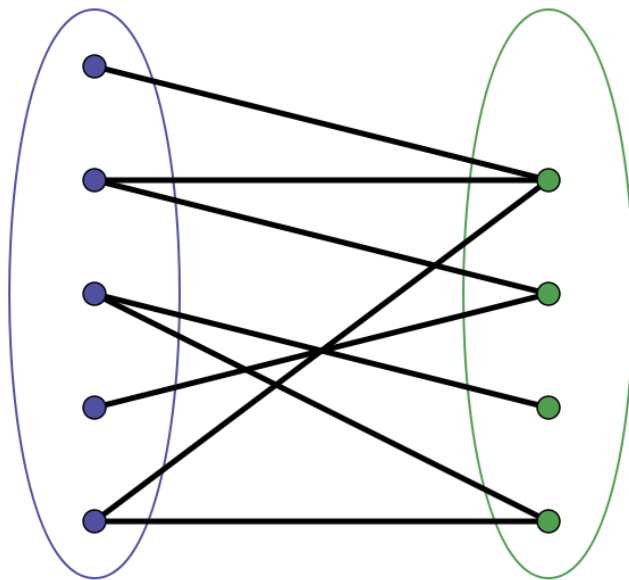
C++ 算法



二分图

Bipartite Graph

二分图定义



a bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets

对于无向图 G ，如所有顶点可分为两个部分，所有边都不出现在每部分内部，则称图 G 为一个二分图

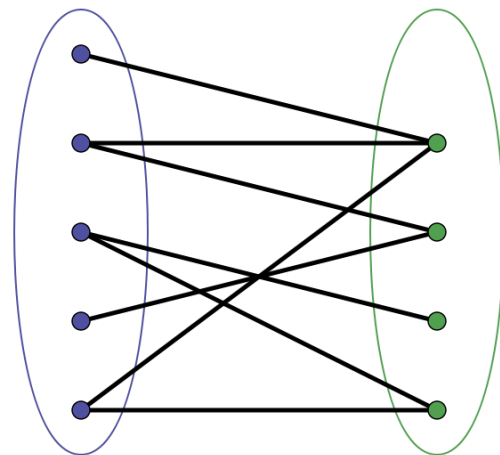
二分图举例

羽毛球混合双打

每个点代表一个球员
每条边代表一个混合双打组合

电视相亲类节目

每个点代表一位嘉宾
每条边代表一对嘉宾配对成功



高智商罪犯五

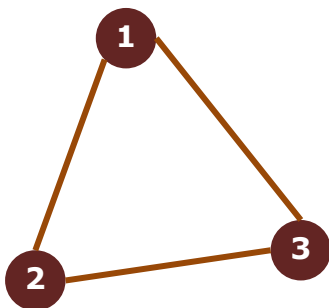
作为监狱长，你的监狱新造了一幢大楼，现在共有两座独立的楼。共 n 个罪犯，编号1到 n 。为了防止高智商罪犯越狱，你希望将一些危险的罪犯重新安排到不同楼里。于是你询问了 m 个狱警，第 i 个狱警列出两个不同的罪犯编号 a_i, b_i ，他们俩不允许被安排在同一个楼里。请问是否有可行方案？

输入第一行为两个正整数 n 和 m 。接下来的 m 行每行正整数 a_i, b_i 。
 $3 \leq n \leq 20000$, $m \leq 50000$. 输出Yes或No

输入样例:

3 3
1 2
2 3
3 1

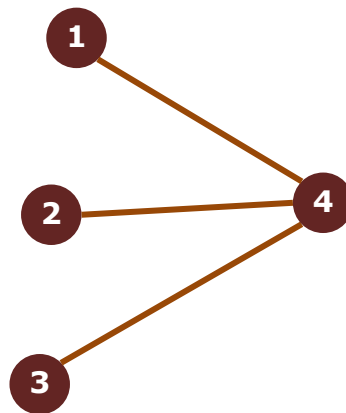
输出样例:
No



输入样例:

4 3
1 4
2 4
3 4

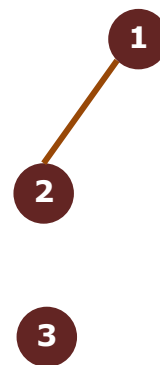
输出样例:
Yes



输入样例:

3 1
1 2

输出样例:
Yes



二分图判定

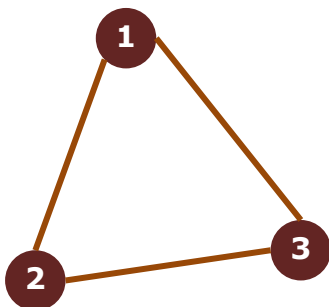
无向图G里共n个节点，编号1到n，共m条无向边。请判定G是不是二分图

输入第一行为两个正整数n和m。接下来的m行每行为正整数 a_i , b_i 。 $3 \leq n \leq 20000$, $m \leq 50000$. 输出Yes或No

输入样例:

3 3
1 2
2 3
3 1

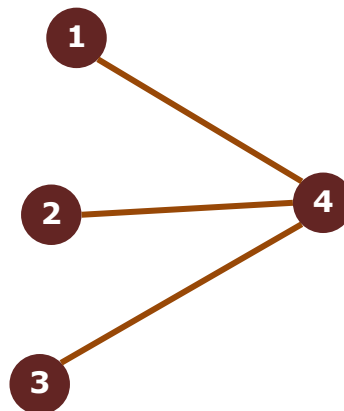
输出样例:
No



输入样例:

4 3
1 4
2 4
3 4

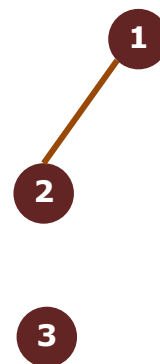
输出样例:
Yes



输入样例:

3 1
1 2

输出样例:
Yes



二分图判定 - 两种算法

1

黑白染色

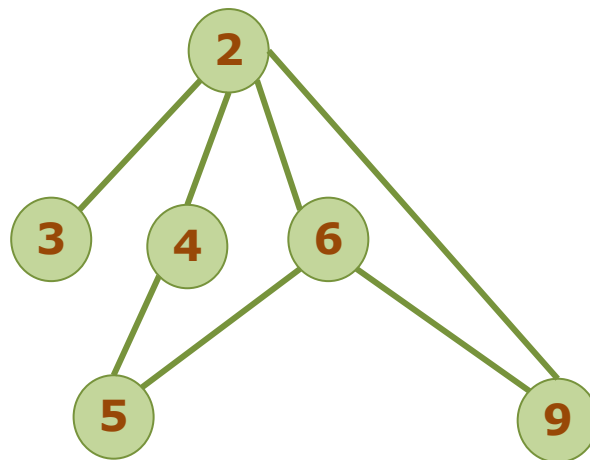
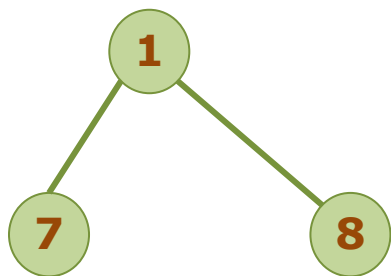
DFS/BFS

2

反物质图

并查集

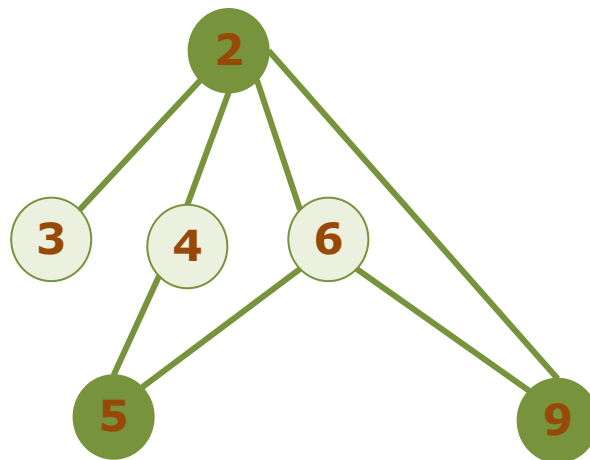
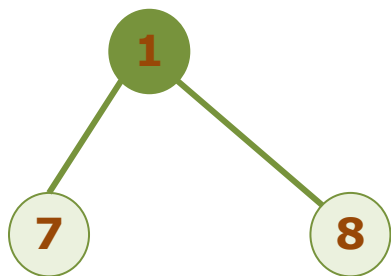
DFS黑白染色



发现(9,2)连边的
两端9,2号同色

不是二分图

DFS黑白染色



发现(9,2)连边的
两端9,2号同色

不是二分图

DFS黑白染色

BiG表示原图是否是
Bipartite Graph

```
4 int n,m,a,b,c,x,BiG,vst[N];
5 vector<int> to[N];
6 void dfs(int u,int color){
7     if(!BiG)return;
8     vst[u]=color;
9     for(int i=0;i<to[u].size();i++)
10         if(!vst[to[u][i]])dfs(to[u][i],3-color);
11         else if(vst[to[u][i]]==color)BiG=0;
12 }
```

请轮流翻译每一行

DFS黑白染色

```
13 int main(){
14     cin>>n>>m;
15     for(int i=0;i<m;i++){
16         cin>>a>>b;
17         to[a].push_back(b);
18         to[b].push_back(a);
19     }
20     fill(vst,vst+n+1,0);
21     BiG=1;
22     for(int i=1;i<=n&&BiG;i++){
23         if(!vst[i]) dfs(i,1);
24     }
25     if(BiG) cout<<"Yes"<<endl;
26     else cout<<"No"<<endl;
27     return 0;
}
```

请轮流翻译每一行

BiG表示原图是否是
Bipartite Graph

代码复习

请用电脑实现
DFS黑白染色
判断二分图
完成“高智商罪犯五”

现场限时20分钟

请附近同学互相找bug

反物质图 - 并查集

输入样例:

4 3

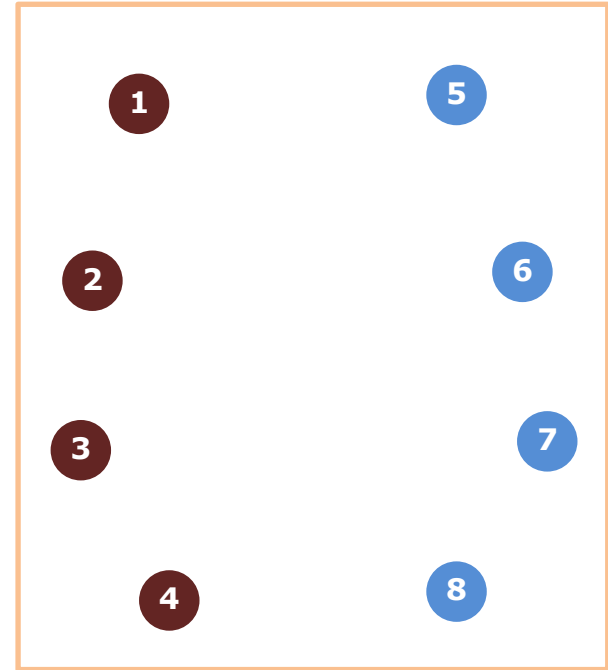
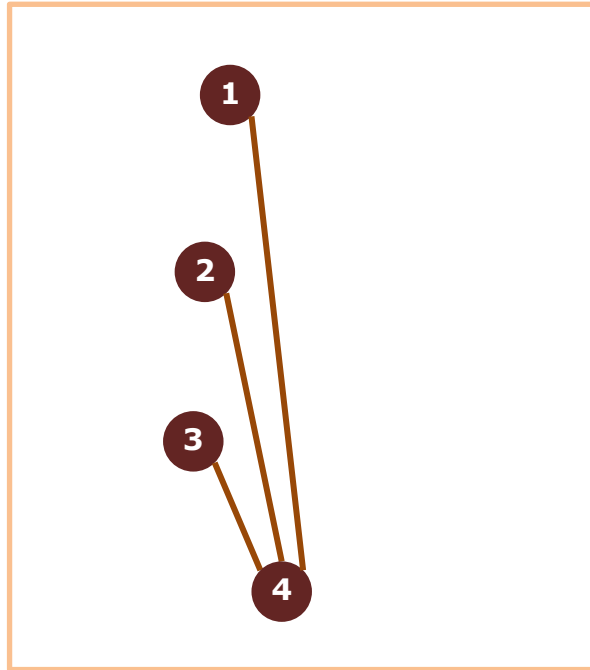
1 4

2 4

3 4

输出样例:

Yes



1号的反面节点记作(1+4)号

2号的反面节点记作(2+4)号

3号的反面节点记作(3+4)号

4号的反面节点记作(4+4)号

反物质图 - 并查集

输入样例:

4 3

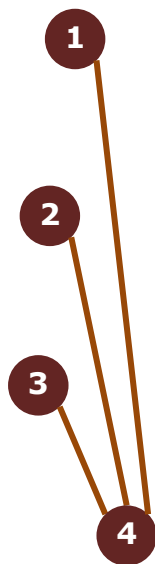
1 4

2 4

3 4

输出样例:

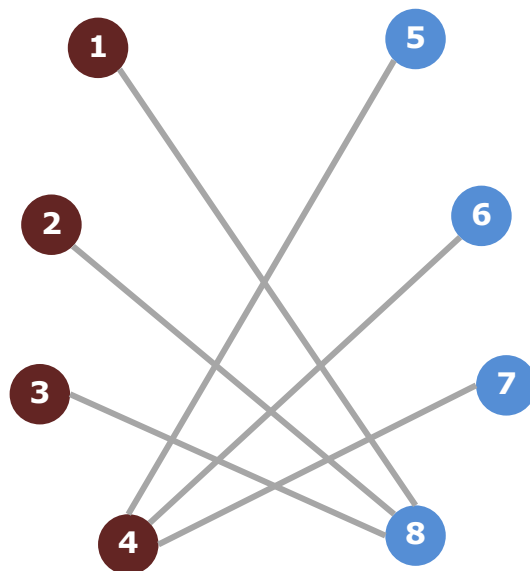
Yes



1和4不可同楼

2和4不可同楼

3和4不可同楼



1和反4(即8号)必同楼

2和反4(即8号)必同楼

3和反4(即8号)必同楼

4和反1(即5号)必同楼

4和反2(即6号)必同楼

4和反3(即7号)必同楼

反物质图 - 并查集

输入样例:

4 4

1 4

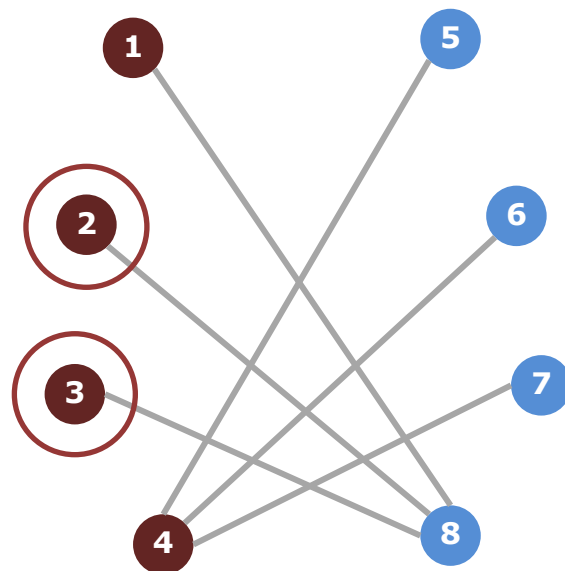
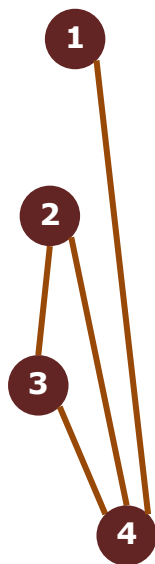
2 4

3 4

2 3

输出样例:

No



1和4不可同楼

2和4不可同楼

3和4不可同楼

2和3不可同楼

1和反4(即8号)必同楼

2和反4(即8号)必同楼

3和反4(即8号)必同楼

发现2和3已经同楼

4和反1(即5号)必同楼

4和反2(即6号)必同楼

4和反3(即7号)必同楼

原图必定不是二分图

错误代码

```
9  cin>>n>>m;
10 for(int i=1;i<=n;i++)id[i]=i;
11 bool BiG=1;
12 for(int i=0;i<m;i++){
13     cin>>a>>b;
14     ll fa=find(a),fb=find(b);
15     if(fa==fb) {BiG=0;break;}
16     unite(a,b+n);
17     unite(b,a+n);
18 }
19 if(BiG)cout<<"Yes"<<endl;
20 else cout<<"No"<<endl;
```

BiG表示原图是否是
Bipartite Graph


```
1 #include<bits/stdc++.h>
2 #define N 40009 ←
3 using namespace std;
4 typedef long long ll;
5 ll n,m,a,b,id[N];
6 ll find(ll x){return id[x]==x?x:id[x]=find(id[x]);}
7 ll unite(ll x,ll y){ll fx=find(x),fy=find(y);id[fx]=fy;}
8 int main(){
9     cin>>n>>m;
10    for(int i=1;i<=n+n;i++)id[i]=i;
11    bool BiG=1;
12    for(int i=0;i<m;i++){
13        cin>>a>>b;
14        ll fa=find(a),fb=find(b);
15        if(fa==fb) {BiG=0;break;}
16        unite(a,b+n); ←
17        unite(b,a+n); ←
18    }
19    if(BiG)cout<<"Yes"<<endl;
20    else cout<<"No"<<endl;
21    return 0;
22 }
```

代码复习

请用电脑实现
并查集造反物质图
判断二分图
完成“高智商罪犯四”

现场限时20分钟

请附近同学互相找bug

DFS-黑白染色

对比

并查集-反物质图

时间 $O(n+m)$

空间 $O(n+m)$

时间约 $O(n+m)$

空间 $O(n)$ 可不存边

读完图后才执行

读图时可以执行

多个连通块难合并

多个连通块可合并

作业

644

算法
讨论

555