

# 图的遍历(染色法判断奇环)

## [传送门](#)

思路：首先图应该为连通图，所需要加的边数即 $add_{edge} = \text{连通块数} - 1$

然后又因为是要每次走两步，即最后要走偶数步即对于走一个奇环。因为奇数条边的环倒退一步就是偶数步。判断是否有奇环用二分图的染色法即可。

如果没有奇环则+1条边即可。

时间复杂度： $O(n + m)$

**AC代码：**

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

using namespace std;

typedef long long ll;

const int N=1e5+5;

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

vector<int>e[N];

int n,m,col[N],ans,ok=1;

void dfs(int u){
    for(auto v:e[u]){
        if(!col[v]){ //染色法
            col[v]=-col[u];
            dfs(v);
        }
        else if(col[v]==col[u]) ok=0; //判断是否有奇环
    }
}

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);
    }
    for(int i=1;i<=n;i++){
        if(!col[i]){
```

```
        ans++; //连通块数
        col[i]=1;
        dfs(i);
    }
}

printf("%d\n",ans-1+ok);
return 0;
}
```