

强连通分量——Tarjan 算法模板

参考代码

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e4+5,M=2e5+5;
int head[N],to[M],nxt[M],tot;
int dfn[N],s[N],low[N],cnt,top;
int sc;// SCC 个数
int scc[N];//结点 i 所在 SCC 的编号
int sz[N];//强连通 i 的大小
bool in[N];
void add(int u,int v){
    nxt[++tot]=head[u];
    head[u]=tot;
    to[tot]=v;
}
void tarjan(int u){
    low[u]=dfn[u]=++cnt;
    s[++top]=u;
    in[u]=1;
    for(int i=head[u];i;i=nxt[i]){
        int v=to[i];
        if(!dfn[v]){
            tarjan(v);
            low[u]=min(low[u],low[v]);
        }
        else if(in[v]){
            low[u]=min(low[u],dfn[v]);
        }
    }
    if(dfn[u]==low[u]){
        ++sc;
        while(s[top]!=u){
            scc[s[top]]=sc;
            sz[sc]++;
            in[s[top]]=0;
            top--;
        }
        scc[s[top]]=sc;
        sz[sc]++;
        in[s[top]]=0;
        top--;
    }
}
int main(){
    int n,m;
```

```
while (scanf("%d %d", &n, &m) && (n | m)) {
    memset(dfn, 0, sizeof(dfn));
    memset(in, 0, sizeof(in));
    memset(s, 0, sizeof(s));
    memset(low, 0, sizeof(low));
    memset(head, 0, sizeof(head));
    memset(to, 0, sizeof(to));
    memset(nxt, 0, sizeof(nxt));
    sc = 0;
    cnt = tot = top = 0;
    for (int i = 1; i <= m; i++) {
        int u, v;
        scanf("%d %d", &u, &v);
        add(u, v);
    }
    for (int i = 1; i <= n; i++) {
        if (!dfn[i]) {
            tarjan(i);
        }
    }
    if (sc == 1) puts("Yes");
    else puts("No");
}
return 0;
}
// hdu 1269 迷宫城堡
```

From:
<https://wiki.cvvbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvvbacm.com/doku.php?id=2020-2021:teams:legal_string:lgwza:%E5%BC%BA%E8%BF%9E%E9%80%9A%E5%88%86%E9%87%8F_tarjan_%E7%AE%97%E6%B3%95%E6%A8%A1%E6%9D%BF

Last update: 2020/08/24 11:32