

问题概述

回路问题指的是有向或无向图中的欧拉回路和哈密顿回路问题。

欧拉回路问题

定义

在一个有向或无向图 G 中，所有边都经过一次且仅经过一次的通路称为欧拉通路，若起点与终点相同，则称为欧拉回路。

若一个图有欧拉回路，则称其为欧拉图，否则若其有欧拉通路，称其为半欧拉图。

判定

无向图欧拉通路的判定条件十分简单，只要图没有奇数度的节点或有且仅有两个奇数度的节点且图是连通图。并且显然若有两个奇数度的节点，那么这两个点必定分别是欧拉通路的起终点。

若无向连通图没有奇数度的节点，那么它一定有欧拉回路。注意这是欧拉回路判定的充要条件，因为每个点都可以进出达到平衡。

有向图的欧拉通路判定条件略为复杂，首先它的基图（即去掉方向的无向图）要连通，其次和无向图对应，所有点出入度相等或一个点出度比入度多一另一个点出度比入度少一、其它点出入度相等。起终点判断显然。

与无向图类似，有向图的欧拉回路判定条件为，基图连通且所有点出入度相等。这同样是充要条件。

这些判断方法的定理和推论都来自离散数学，不过多赘述其实是我不会。

求解

方法一 DFS求解

这一方法非常显然，找到一个正确的起点开始DFS，走不通了就回溯，在栈中记录下经过的边的顺序，当走完所有边时结束算法即可。

```
#include <map>
#include <set>
#include <ctime>
#include <cmath>
#include <stack>
#include <queue>
#include <vector>
#include <cstdio>
#include <cstdlib>
```

```
#include <cstring>
#include <iostream>
#include <algorithm>
using namespace std;

const int maxn = 10000 + 5;
const int maxm = 100000 + 5;

int N;//点数
int M;//边数
int in[maxn], out[maxn]; //点的出入度
int flag;//是否找到答案
int F[maxn], nxt[maxm << 1], v[maxm << 1], ban[maxm << 1], EID = 0;//邻接表,
注意因为要ban掉反边所以从0开始编号
int st;//起点
int stk[maxm], top = 0;//记录路径

inline void add(int f, int t)
{
    nxt[EID] = F[f];
    v[EID] = t;
    F[f] = EID++;
}

inline void dfs(int x)
{
    stk[++top] = x;
    if(top == M + 1)
    {
        for(int i = 1; i <= top; ++i)
            printf("%d ", stk[i]);
        printf("\n");
        flag = true;
        return;
    }
    for(int i = F[x]; i != -1; i = nxt[i])
    {
        if(ban[i])
            continue;
        ban[i] = 1;
        ban[i ^ 1] = 1;
        dfs(v[i]);
        ban[i] = 0;
        ban[i ^ 1] = 0;
        if(flag)
            return;
    }

    --top;
}
```

```

int main()
{
    memset(F, -1, sizeof F);
    scanf("%d%d", &N, &M);
    for(int i = 1; i <= M; ++i)
    {
        int u, v;
        scanf("%d%d", &u, &v);
        add(u, v);
        add(v, u);
        ++in[v], ++out[u];
    }

    for(int i = 1; i <= N; ++i) // 找到起点
        if(out[i] & 1)
        {
            st = i;
            break;
        }
    if(st == 0) // 若全都出度等于入度说明有欧拉回路，可以从任意一个点开始
        st = 1;

    dfs(st);

    return 0;
}

```

方法二 Fleury算法

设 G 为一无向欧拉图，求 G 中的一个欧拉回路。

(1) 首先，任取 G 中的一个顶点 V_0 ，令 $P_0 = V_0$

(2) 假设沿 $P_i = V_0 e_1 V_1 e_2 V_2 \dots e_i V_i$ 走到顶点 V_i ，按照下面的方法从集合 $E(G) - \{e_1, e_2, \dots, e_i\}$ 中选取 e_{i+1}

(a) e_{i+1} 与 V_i 相关联

(b) 除非没有边可以选择，否则 e_{i+1} 不是 $G - \{e_1, e_2, \dots, e_i\}$ 中的桥。

(3) 当(2)不能继续进行时结束算法。

From:

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

Permanent link:

<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:hotpot:circuit&rev=1589978788>

Last update: 2020/05/20 20:46