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$P_i=V_0e_1V_1e_2V_2 \ldots e_iV_i$走到顶点$V_i$，按照下面的方法从集合$E(G)-\{e_1,e_2 \ldots e_i\}$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$G-\{e_1,e_2 \ldots e_i\}$中的桥。

(3)当(2)不能继续进行时结束算法。

~~先研究一下边表写法~~代码看这篇博客

哈密顿回路问题

定义

哈密顿回路指的是一个回路通过图的每个点一次且仅一次，每条边最多一次的回路。

判定

Dirac定理(充分条件)

设一个无向图中有N个顶点，若所有顶点的度数大于等于$\frac{N}{2}$，则哈密顿回路一定存在，($\frac{N}{2}$指的是$[\frac{N}{2}]$，向上取整)。

必要条件

设图$G=\langle V,E \rangle$是哈密顿图，则对于V的任意一个非空子集S，若以$|S|$表示S中元素的数目，$G-S$表示G中删除了S中的点以及这些点所关联的边后得到的子图，则$W(G-S) \le |S|$成立。其中$W(G-S)$是$G-S$中连通分量数。

$N(N \ge 2)$阶竞赛图一定有哈密顿回路。（竞赛图是通过在无向完整图中为每个边分配方向而获得的有向图）

求解

由Dirac定理为前提构造

(1) 任意找两个相邻的节点S和T，在其基础上扩展出一条尽量长的没有重复结点的路径。即如果S与结点v相邻,而且v不在路径$S \rightarrow T$上,则可以把该路径变成$v \rightarrow S \rightarrow T$，然后$v$成为新的$S$，从$S$和$T$分别向两头扩展，直到无法继续扩展为止，即所有与$S$或$T$相邻的节点都在路径$S \rightarrow T$上。

(2) 若S与T相邻，则路径$S \rightarrow T$形成了一个回路。

(3) 若S与T不相邻,可以构造出来一个回路。设路径$S \rightarrow T$上有$k + 2$个节点，依次为$S, v1, v2 \ldots vk, T$，可以证明存在节点$v_i(i \in [1, k])$，满足v_i与T相邻，且v_{i+1}与S相邻。找到这个节点v_i，把原路径变成$S \rightarrow v_i \rightarrow T \rightarrow v_{i+1} \rightarrow S$，即形成了一个回路。

(4) 到此为止，已经构造出来了一个没有重复节点的的回路，如果其长度为N，则哈密顿回路就找到了。如果回路的长度小于N，由于整个图是连通的，所以在该回路上，一定存在一点与回路之外的点相邻。那么从该点处把回路断开，就变回了一条路径，同时还可以将与之相邻的点加入路径。再按照步骤1的方法尽量扩展路径，则一定有新的节点被加进来。接着回到步骤(2)。