

欧拉回路/通路

Euler Circuit/Path

Concept: 图 G 的欧拉回路是包含 G 中每一条边的简单回路；图 G 的欧拉通路是包含 G 中每一条边的简单通路。（简单：不重复走边，即经过所有边恰好一次）

Theorem:

- 无向图存在欧拉回路当且仅当该图没有奇度顶点；
- 无向图存在欧拉通路当且仅当该图有且仅有 2 个奇度顶点；
- 有向图存在欧拉回路当且仅当该图所有点出度等于入度；
- 有向图存在欧拉通路当且仅当该图仅有 2 个点出度不等于入度，且这两个点其一入度比出度大 1，另一出度比入度大 1。

Heirholzer Algorithm

Idea: dfs 一遍，每次回溯时把当前点加入答案；无需打 `vis` 标记，不断删边即可。

Complexity: $O(V + E)$

Code (`vector` 存边) :

```
1  vector<pii> edge[N];
2  // edge[i].first is the vertex;
3  // edge[i].second is the id of this edge
4
5  bool vis[M];
6  stack<int> sta;
7  void dfs(int x){
8      while(!edge[x].empty()){
9          pii to = edge[x].back(); edge[x].pop_back();
10         if(vis[to.second]) continue;
11         vis[to.second] = true; // mark the edge
12         dfs(to.first);
13     }
14     sta.push(x);
15 }
16 int main(){
17     read(n, m);
```

```
18     for(int i = 1; i <= m; i++){
19         int u, v; read(u, v);
20         edge[u].pb(mp(v, i));
21         edge[v].pb(mp(u, i));
22     }
23     int st = 1;
24     for(int i = 1; i <= n; i++){
25         if(edge[i].size() & 1){
26             st = i;
27             break;
28         }
29     }
30     dfs(st);
31     while(!sta.empty()){
32         printf("%d\n", sta.top());
33         sta.pop();
34     }
35     return 0;
36 }
```