

使用深度优先搜索 (DFS)。通过跟踪结点状态（未访问、正在访问、已访问）和父结点，若 DFS 遇到一个非父结点的”正在访问”状态的邻居，则发现环。此时，沿父结点链回溯可得环路径。 $O(n + m)$ 。每个结点和每条边仅访问常数次。

Algorithm 1 无向图环检测算法

```

1: function DETECTANDPRINTCYCLE( $G = (V, E), n$ )
2:   state: 数组，初始全为 0 (未访问)
3:   parent: 数组，初始全为 -1
4:   cyclePath: 空列表
5:   for 每个结点  $u$  从 0 到  $n - 1$  do
6:     if  $state[u] = 0$  then ▷ 若未访问
7:       if DFSFINDCYCLE( $u, G, state, parent, cyclePath$ ) then
8:         return  $cyclePath$  ▷ 发现环
9:       end if
10:    end if
11:  end for
12:  return  $\emptyset$  ▷ 无环
13: end function

14: function DFSFINDCYCLE( $u, G, state, parent, cyclePath$ )
15:    $state[u] \leftarrow 1$  ▷ 正在访问
16:   for 每个邻居  $v$  在  $G[u]$  中 do
17:     if  $state[v] = 1$  then ▷ 若  $v$  正在访问
18:       if  $v \neq parent[u]$  then ▷ 且  $v$  非父结点，则构成环
19:          $cyclePath.Add(v)$ 
20:          $curr \leftarrow u$ 
21:         while  $curr \neq v$  do
22:            $cyclePath.Add(curr)$ 
23:            $curr \leftarrow parent[curr]$ 
24:         end while
25:          $REVERSE(cyclePath)$ 
26:         return True
27:       end if
28:     else if  $state[v] = 0$  then ▷ 若  $v$  未访问
29:        $parent[v] \leftarrow u$ 
30:       if DFSFINDCYCLE( $v, G, state, parent, cyclePath$ ) then
31:         return True
32:       end if
33:     end if
34:   end for
35:    $state[u] \leftarrow 2$  ▷ 已访问 return False
36: end function
  
```
