

算法作业 9

孟妍廷 2015202009

2017 年 11 月 29 日

22.2-8

分析：首先一棵树应该是无向的，对于树 T ，其节点信息应该有如下两种情况：1. 直接知道哪个结点是叶结点，哪个是根结点 2. 只知道各个节点的邻接关系。对于两种情况设计算法如下：

1. 直接知道根结点

假设根结点为 s

```
solution1( $T, s$ )
   $BFS(T, s)$ 
   $D, d = -\infty$  //  $D$  是一个用于记录的空结点
  for each  $v \in T.Adj[s]$ 
    if  $v.d > D.d$ 
       $D = v$ 
  return  $D$ 
```

时间复杂度为 $O(|V| + |E|)$

2. 只知道邻接关系

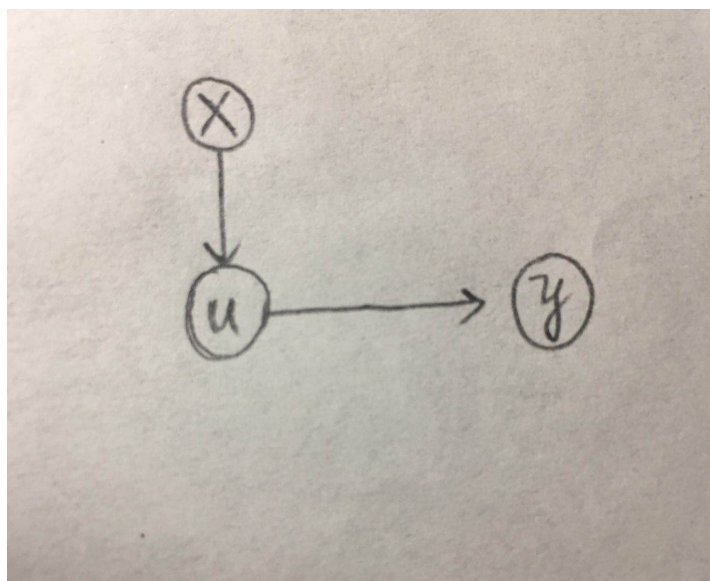
借助 1 中的算法 solution1

```
solution2( $T$ )
  // 任意选取树中的一个结点  $u$ 
   $s = solution1(T, u)$  // 先找到任意一个结点  $u$  的最短路径距离最长的结点  $s$ ,  $s$  一定为直径的一端
   $D = solution1(T, s)$  // 以端点开始执行  $BFS$  找最短路径最长结点即为目标
  return  $D$ 
```

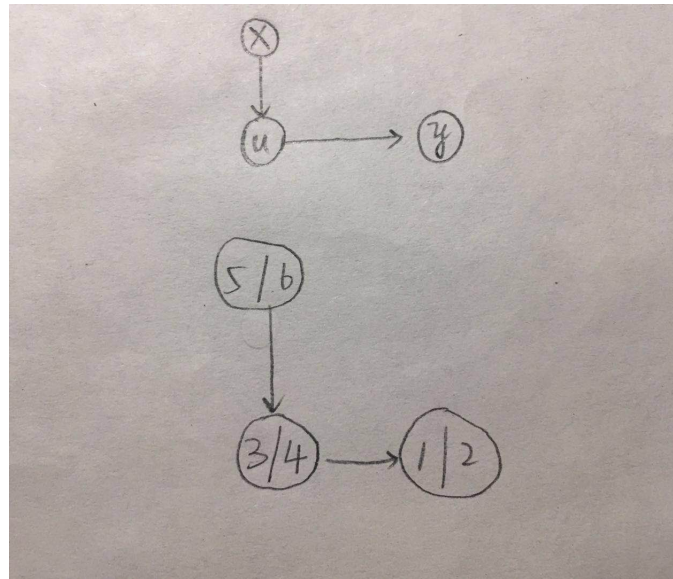
时间复杂度为 $O(|V| + |E|)$

22.3-11

分析：首先唯一结点的意思是指该节点没有树边与其他结点连接，与选择节点的顺序造成的开始时间和完成时间有关。考虑具有如下性质的结点 u



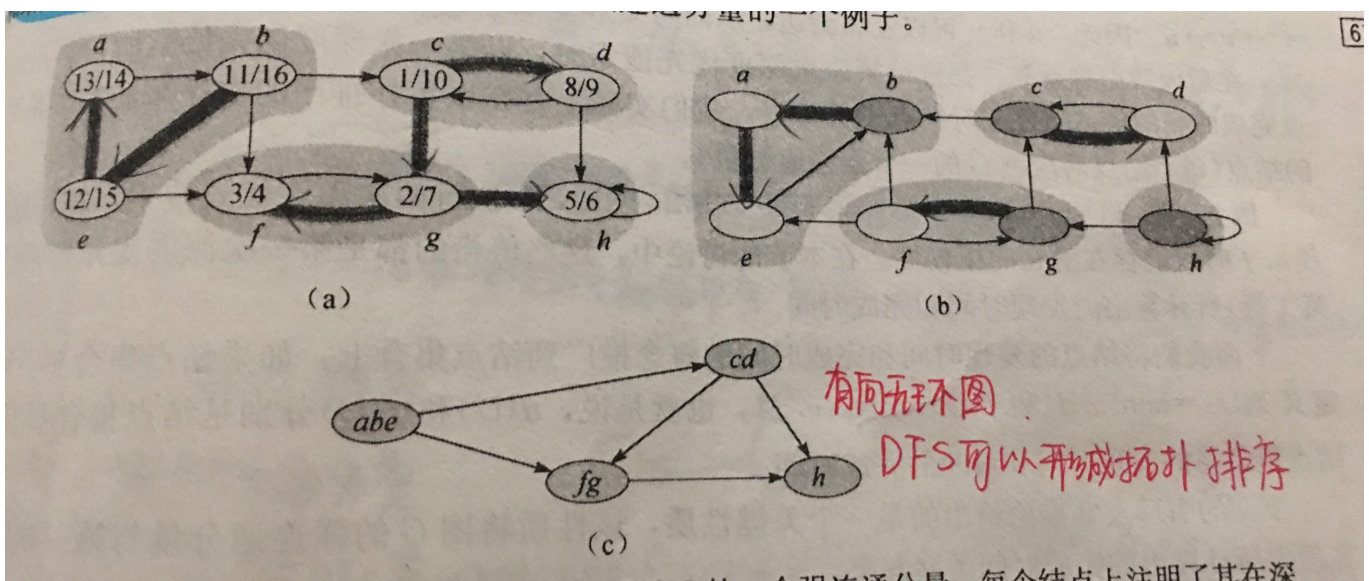
按照一定顺序选取结点进行深度优先算法，可得各结点的开始时间和结束时间如下：



也就是说，对于既有出边也有入边的结点 u ，若进行深度优先算法使其出边指向的结点在它被选择之前就被选择，而指向它的结点在它之后被选择就会使其成为唯一节点
因此具有这样的结构的既有出边也有入边的结点 u 能成为深度优先树中的唯一节点。

22.5-3

解：不一定能计算出结果。以书上的例子作为反例



若不转置，从结束时间最短的结点开始正序遍历，则第一个选择的结点为 f ， f 指向 g ，此时 g 既可以指向 f 也可以指向 h ，产生的冲突，并不能保证一定能形成强连通图。