数据结构与算法（Python版）

通用的深度优先搜索

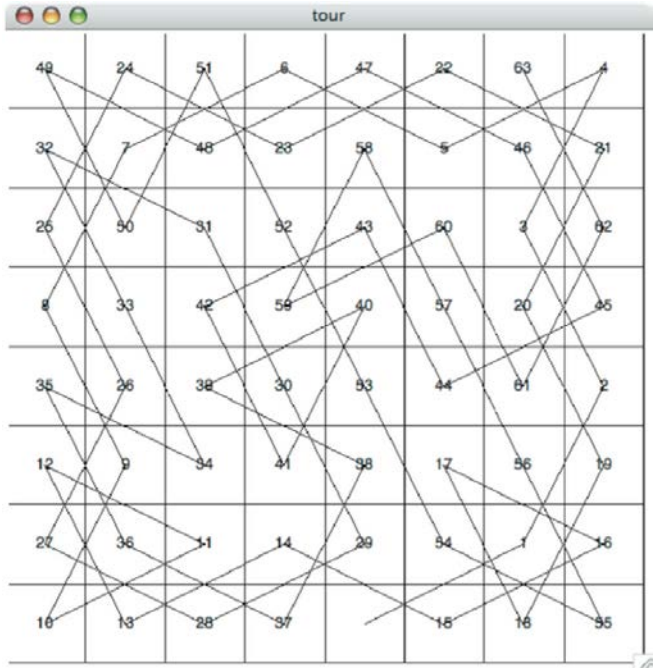
陈斌 北京大学 gischen@pku.edu.cn

通用的深度优先搜索

- 骑士周游问题是一种特殊的对图进行深度优先搜索

  其目的是建立一个没有分支的最深的深度优先树

  表现为一条线性的包含所有节点的退化树



通用的深度优先搜索

- 一般的深度优先搜索目标是在图上进行尽量深的搜索，连接尽量多的顶点，必要时可以进行分支（创建了树）

  有时候深度优先搜索会创建多棵树，称为“深度优先森林”

- 深度优先搜索同样要用到顶点的“前驱”属性，来构建树或森林

  另外要设置“发现时间”和“结束时间”属性

  - 前者是在第几步访问到这个顶点（设置灰色）
  - 后者是在第几步完成了此顶点探索（设置黑色）

  这两个新属性对后面的图算法很重要

通用的深度优先搜索

- 带有DFS算法的图实现为Graph的子类
  - 顶点Vertex增加了成员Discovery及Finish
  - 图Graph增加了成员time用于记录算法执行的步骤数目

通用的深度优先搜索算法代码

BFS采用队列存储待访问顶点
DFS则是通过递归调用，隐式使用了栈

颜色初始化

如果还有未包括的顶点，则建森林

算法的步数

深度优先递归访问

```
from pythonds.graphs import Graph
class DFSGraph(Graph):
    def __init__(self):
        super().__init__()
        self.time = 0

    def dfs(self):
        for aVertex in self:
            aVertex.setColor('white')
            aVertex.setPred(-1)
        for aVertex in self:
            if aVertex.getColor() == 'white':
                self.dfsvisit(aVertex)

    def dfsvisit(self,startVertex):
        startVertex.setColor('gray')
        self.time += 1
        startVertex.setDiscovery(self.time)
        for nextVertex in startVertex.getConnections():
            if nextVertex.getColor() == 'white':
                nextVertex.setPred(startVertex)
                self.dfsvisit(nextVertex)
        startVertex.setColor('black')
        self.time += 1
        startVertex.setFinish(self.time)
```

通用的深度优先搜索算法：示例

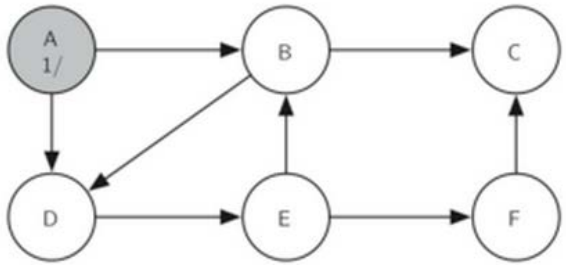


Figure 14: Constructing the Depth First Search Tree-10

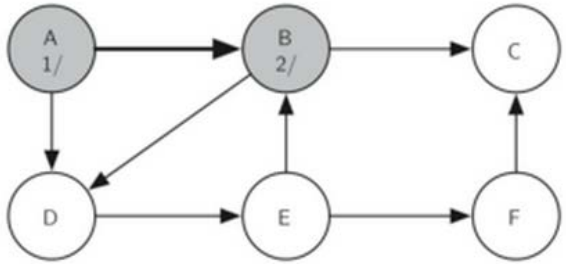


Figure 15: Constructing the Depth First Search Tree-11

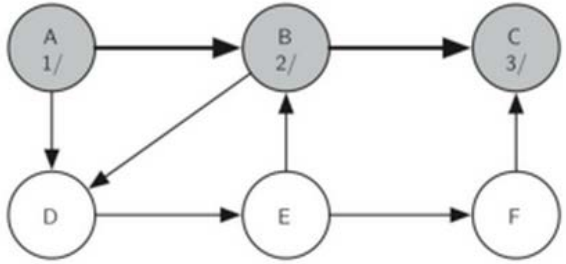


Figure 16: Constructing the Depth First Search Tree-12

通用的深度优先搜索算法：示例

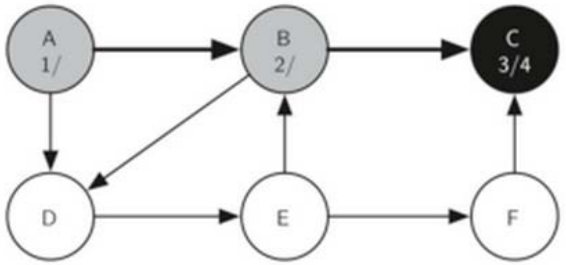


Figure 17: Constructing the Depth First Search Tree-13

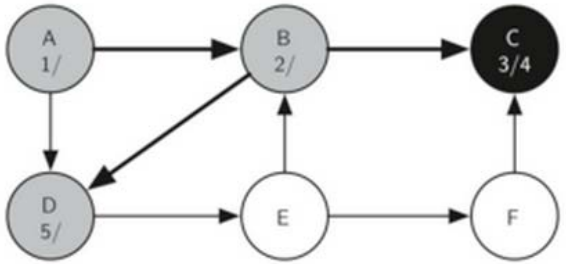


Figure 18: Constructing the Depth First Search Tree-14

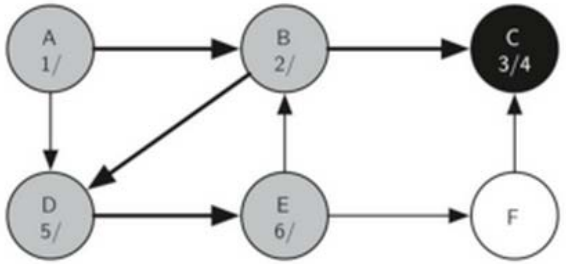


Figure 19: Constructing the Depth First Search Tree-15

通用的深度优先搜索算法：示例

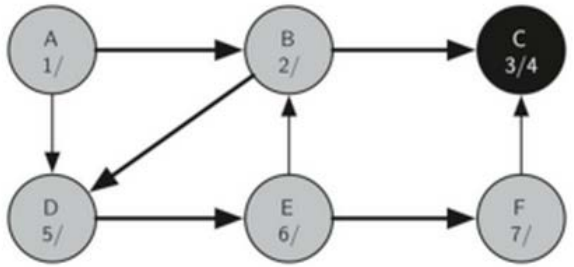


Figure 20: Constructing the Depth First Search Tree-16

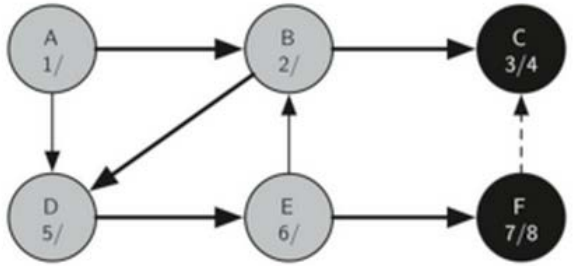


Figure 21: Constructing the Depth First Search Tree-17

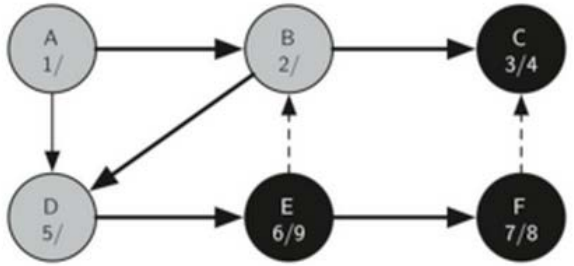


Figure 22: Constructing the Depth First Search Tree-18

通用的深度优先搜索算法：示例

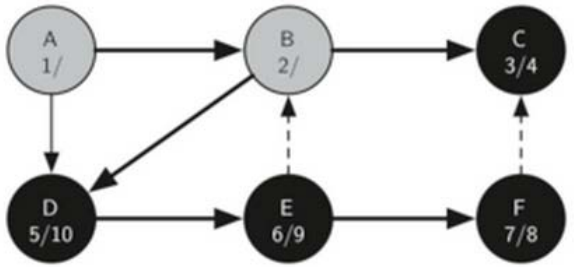


Figure 23: Constructing the Depth First Search Tree-19

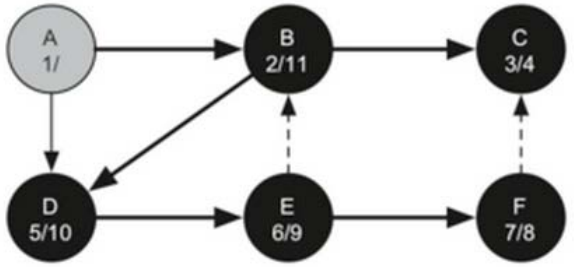


Figure 24: Constructing the Depth First Search Tree-20

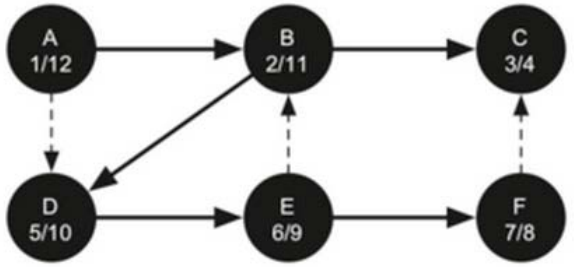


Figure 25: Constructing the Depth First Search Tree-21

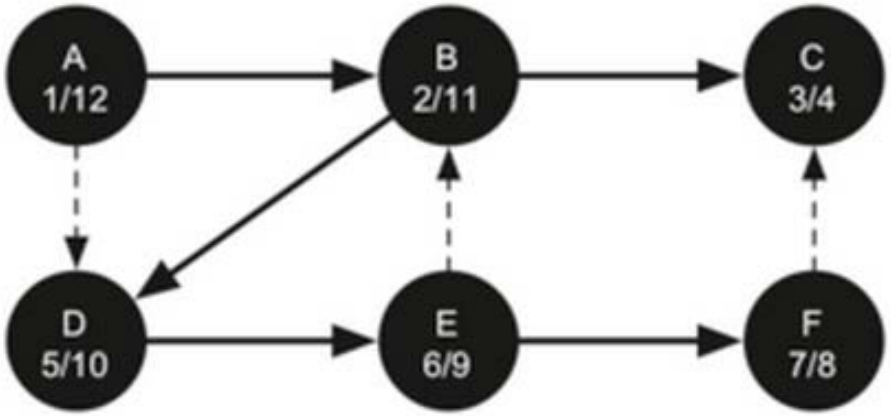
通用的深度优先搜索算法：分析

- DFS构建的树，其顶点的“发现时间”和“结束时间”属性，具有类似**括号**的性质

  即一个顶点的“发现时间”总小于所有子顶点的“发现时间”

  而“结束时间”则大于所有子顶点“结束时间”

  比子顶点更早被发现，更晚被结束探索



通用的深度优先搜索算法：分析

- DFS运行时间同样也包括了两方面：

  dfs函数中有两个循环，每个都是|V|次，所以是O(|V|)

  而dfsvisit函数中的循环则是对当前顶点所连接的顶点进行，而且仅有在顶点为白色的情况下才进行递归调用，所以对每条边来说只会运行一步，所以是O(|E|)

  加起来就是和BFS一样的O(|V|+|E|)