DEPTH FIRST SEARCH

Section Overview

- Overview of General Depth First Search
- Afterwards implementation walkthroughs
  - Graphs
  - BFS
  - DFS

DFS

- The knight’s tour is a special case of a depth first search where the goal is to create the deepest depth first tree, without any branches.
- The more general depth first search is actually easier.
- Its goal is to search as deeply as possible, connecting as many nodes in the graph as possible and branching where necessary.

DFS

- The knight’s tour is a special case of a depth first search where the goal is to create the deepest depth first tree, without any branches.
- The more general depth first search is actually easier.
- Its goal is to search as deeply as possible, connecting as many nodes in the graph as possible and branching where necessary.

DFS

- It is even possible that a depth first search will create more than one tree.
- When the depth first search algorithm creates a group of trees we call this a **depth first forest**.
- As with the breadth first search our depth first search makes use of predecessor links to construct the tree.

DFS

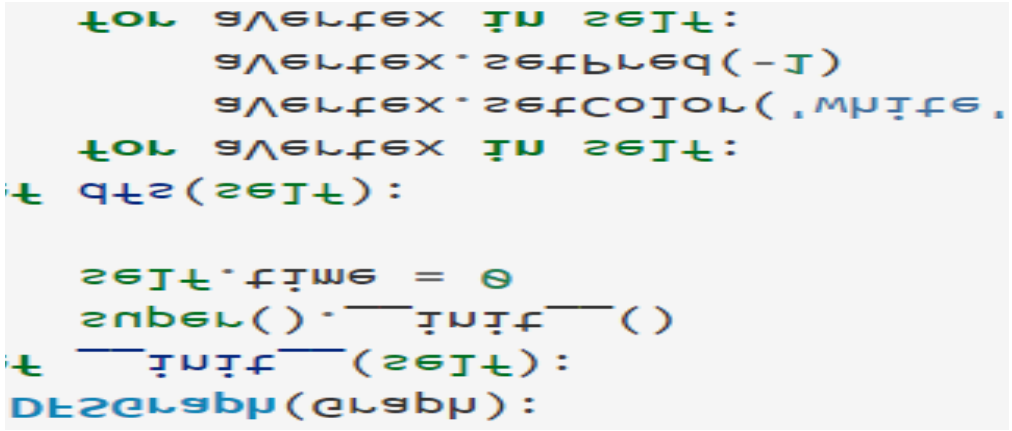
- As with the breadth first search our depth first search makes use of predecessor links to construct the tree.
- In addition, the depth first search will make use of two additional instance variables in the Vertex class.
- The new instance variables are the discovery and finish times.
- The discovery time tracks the number of steps in the algorithm before a vertex is first encountered.
- The finish time is the number of steps in the algorithm before a vertex is colored black

DFS

```
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)
```



DFS

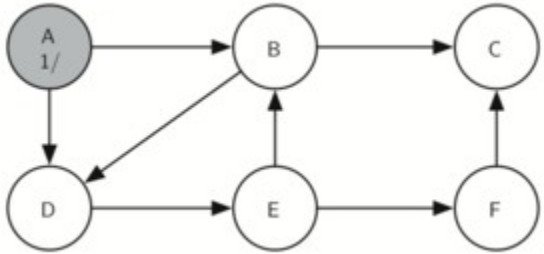
```
class DFSGraph(Graph):
    def __init__(self):
        super().__init__()
        self.time = 0

    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)
```

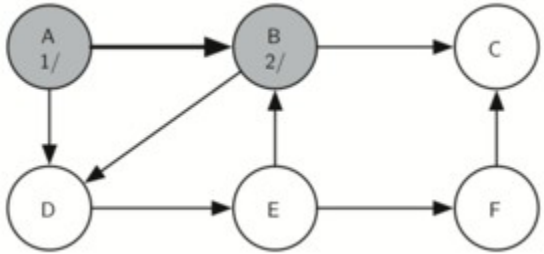
DFS

- Constructing the Depth First Search Tree



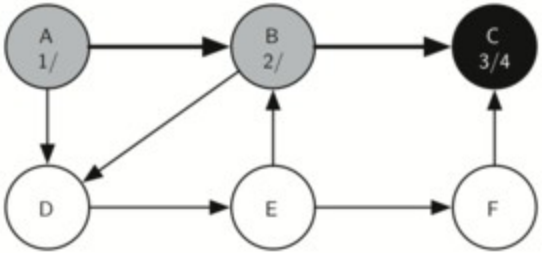
DFS

- Constructing the Depth First Search Tree



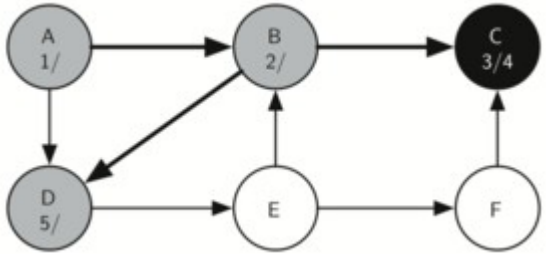
DFS

- Constructing the Depth First Search Tree



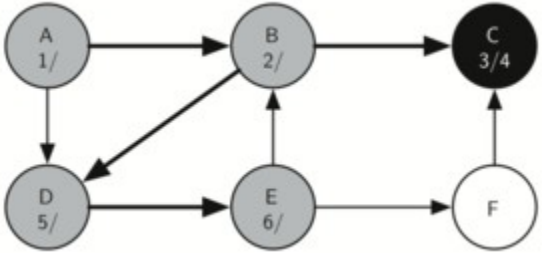
DFS

- Constructing the Depth First Search Tree



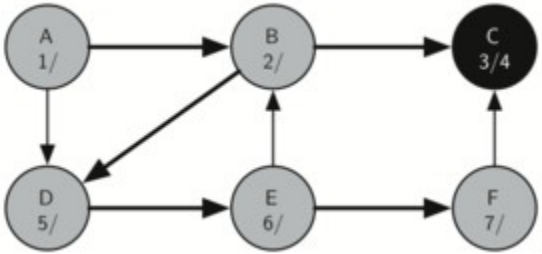
DFS

- Constructing the Depth First Search Tree



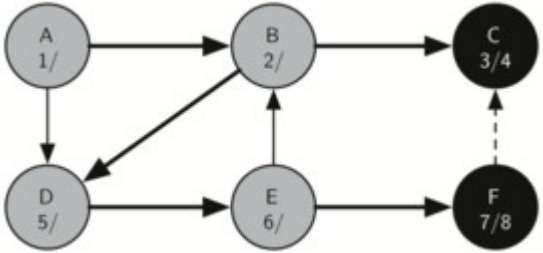
DFS

- Constructing the Depth First Search Tree



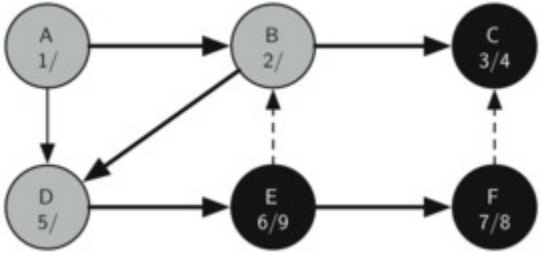
DFS

- Constructing the Depth First Search Tree



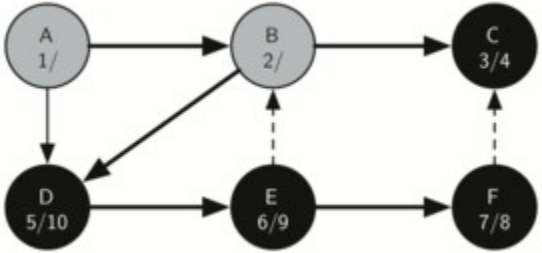
DFS

- Constructing the Depth First Search Tree



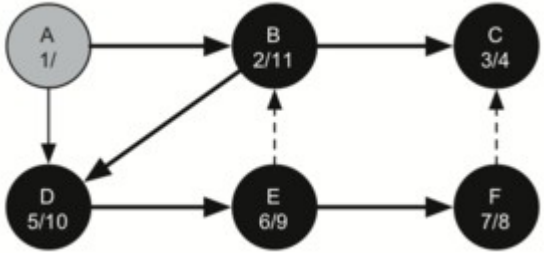
DFS

- Constructing the Depth First Search Tree



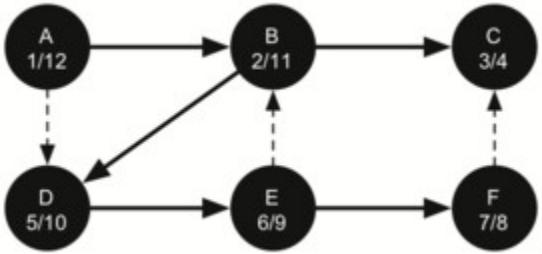
DFS

- Constructing the Depth First Search Tree



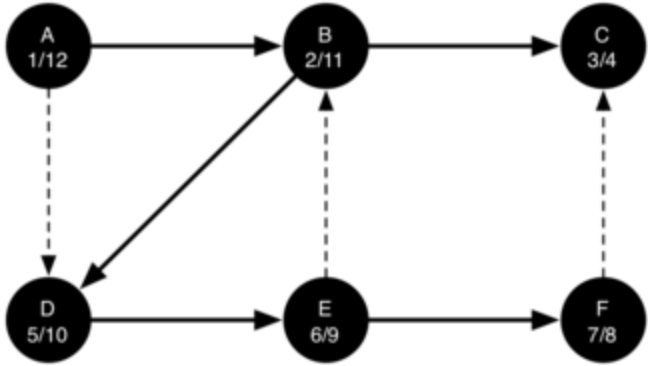
DFS

- Constructing the Depth First Search Tree



DFS

- The starting and finishing times for each node display a property called the **parenthesis property**.
- This property means that all the children of a particular node in the depth first tree have a later discovery time and an earlier finish time than their parent.



DFS

- Up next…Live Code Implementation Walkthroughs
  - Graph and Nodes
  - Breadth First Search
  - Depth First Search