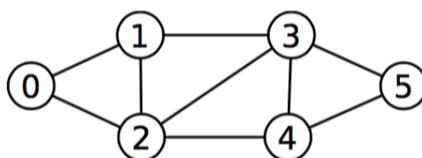


Massachusetts Institute of Technology  
 1.200J—Transportation Systems Analysis: Performance and Optimization  
 Fall 2015 — TA: Wichinpong “Park” Sinchaisri

**Recitation 9**  
**Unit 4 — Network Flows: Models and Algorithms**

## 1 Search in Graph



### 1.1 BFS vs DFS

- (a) Run BFS on the graph in Figure 1. When we had a choice, push the lowest numbered neighbor first. Draw the resulting BFS search tree.
- (b) Re-do BFS on the same graph, but this time, push the highest numbered neighbor first. Draw the resulting BFS search tree. Is it the same as the one in (a)?
- (c) Re-do (a) and (b) but with DFS instead.

### 1.2 Depth First Search

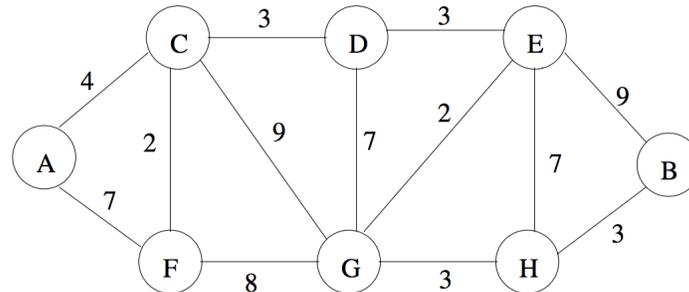
$$A : \{C, E, B\}, \quad B : \{E, D\}, \quad C : \{E\}, \quad D : \{C, F, E\}, \quad E : \{F\}, \quad F : \emptyset$$

- (a) Draw the DFS tree of the graph described by the adjacency list above that will result if the first call is vertex  $A$ . Label each vertex with its pre(in)- and post(out)-order numbers. Assume that DFS procedure visits the vertices in alphabetical order. Draw tree edges as solid lines, and back, cross, and forward edges as dashed lines. Denote them as ‘B’, ‘C’, or ‘F’ on each dashed line.
- (b) Suppose this graph is a precedence graph. Either give a valid order in which to perform the tasks (call them task A, task B, ..., task F) or prove that there is no valid order.

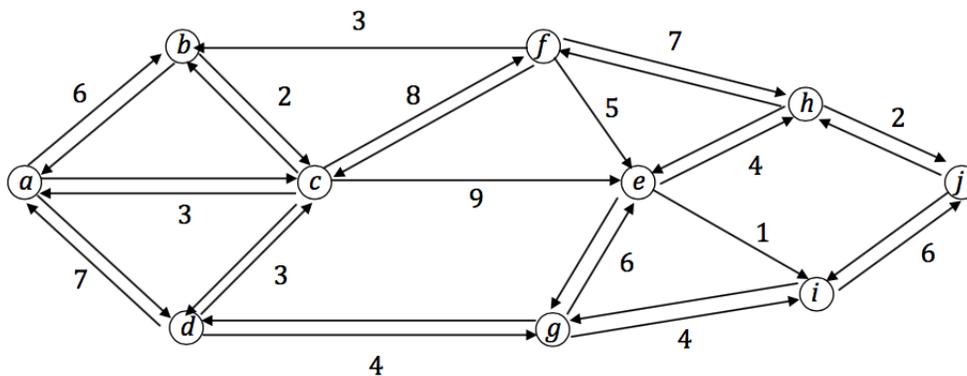
**1.3 Wrestler’s Rivalry** Between any pair of professional wrestlers, there may or may not be a rivalry. For marketing purposes, we want to divide wrestlers in two separate groups: “good guys” and “bad guys.” Suppose we have  $n$  wrestlers and a list of  $m$  rivalries between pairs of wrestlers. Describe an algorithm that runs in  $O(n+m)$  time and determines whether it is possible to split the wrestlers into the two groups, so that each rivalry is between a good guy and a bad guy.

## 2 Shortest Path Problems

**2.1 Dijkstra's Algorithm** A region contains a number of towns connected by roads. Each road is labeled by the average number of minutes required for a fire engine to travel it. Each intersection is labeled with a circle. Suppose that you have decided to place your fire station at location  $G$ .



- What algorithm would you recommend be used to find the fastest route from the fire station to each of the intersections?
- Suppose one location must be selected such that we minimize the distance to the farthest intersection. Devise an algorithm to solve this problem given any arbitrary road map. Analyze its time complexity when there are  $N$  nodes and  $R$  roads.



- How to find the fastest route from all intersections to location  $a$ ?
- If we only know BFS but not Dijkstra's, how to find the fastest route in general?

**2.2 Traffic Planning** Park is planning to drive across a city, and has represented the road network as a graph. Each edge has a weight  $c(e, t)$  describing how much fuel he'll burn idling at badly-timed traffic lights and honking at awful drivers.  $c$  is also a function of time because some roads are more congested at different times during the day. He needs to arrive at some point in the next 12 hours, but beyond that, his objective is to minimize fuel consumption. How should he solve this problem using the algorithm we know?