Habib University

CS 102– Data Structures & Algorithm
Spring 2021

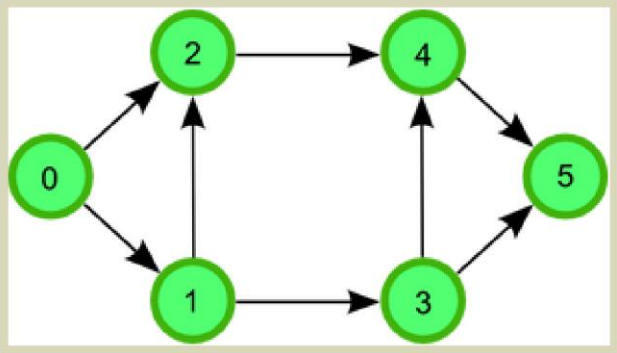
Lab# 11

Graph Traversal

Objectives: In this lab, we will implement graph traversal algorithms we have already studied in the lectures. We will further use graphs to solve some practical problems.

Exercise # 1: Depth First Search

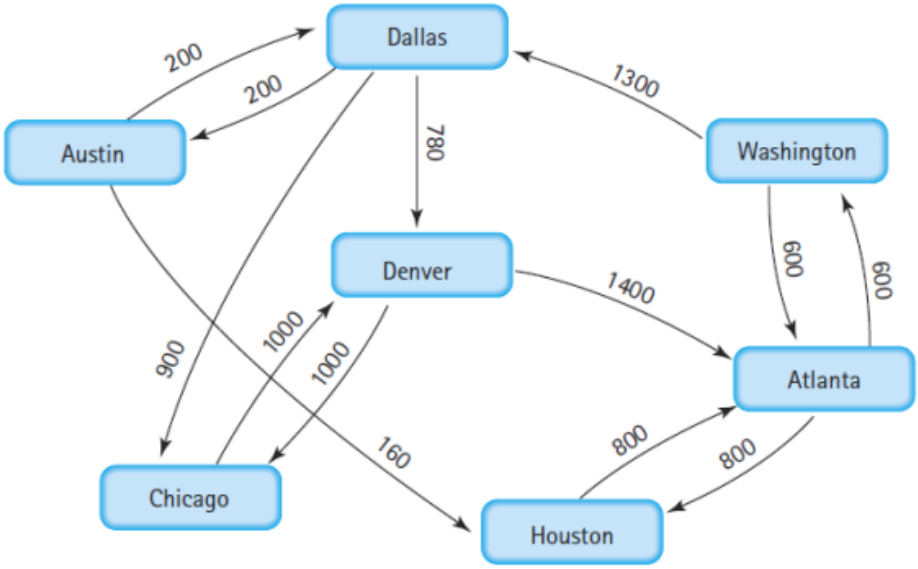
Implement the Depth-first Search (DFS) algorithm and test it on the graph shown below.



Example 1 :	
Input	Adjacency list representation of graph shown above. Source Vertex
Output	List of visited vertices

Exercise # 2 – Detecting Cycle between List of N Airports

The following graph is an example from Rosen (2011). It shows the flights and distances between some of the major airports in the United States.



Write a function that takes a list of N airports and checks if they form a cycle of size N.

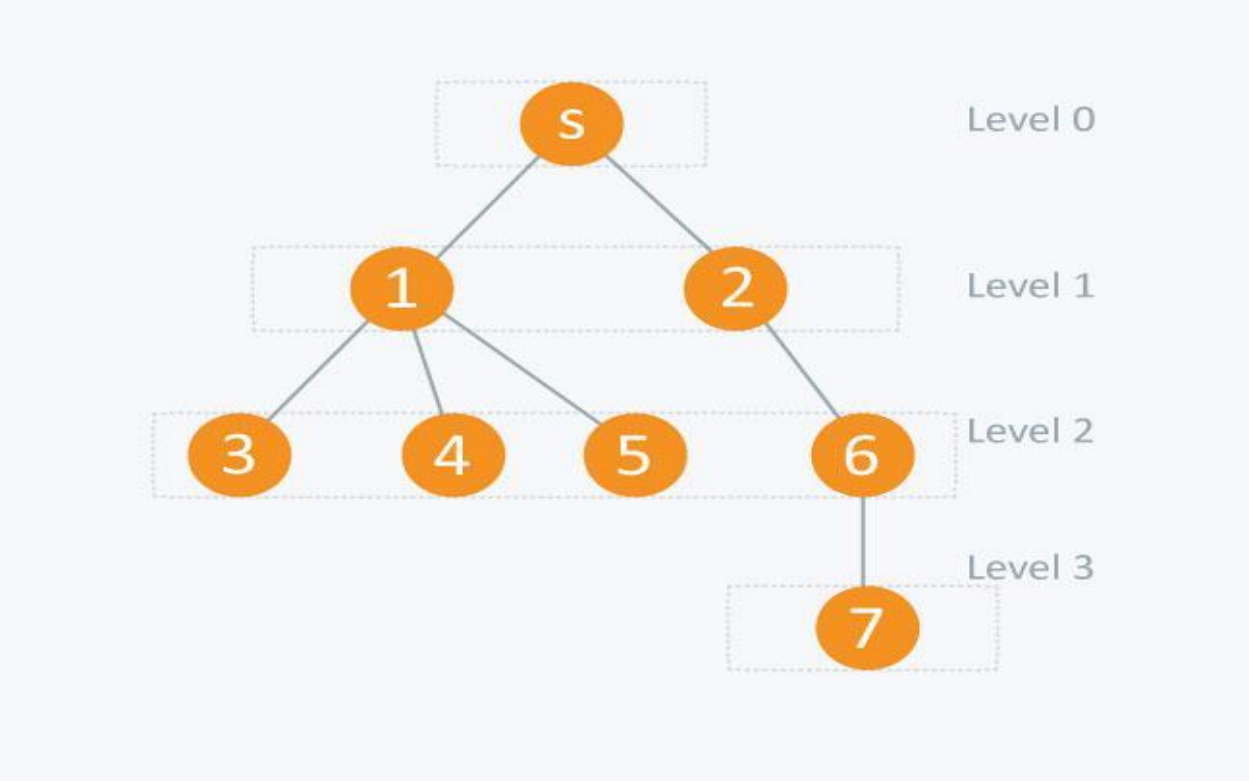
{A cycle is a directed path that starts and ends at the same vertex. Before writing code, make sure you can identify cycles yourself}

```
>>> check_cycles(G, ['Austin','Houston','Atlanta','Washington','Dallas'])
Yes

>>> check_cycles(G, ['Austin','Houston','Atlanta','Washington'])
No
```

Exercise # 3: Breadth First Search

Implement the Breadth-First Search (BFS) algorithm (traversing left to right in each level) and test it on the graph below.

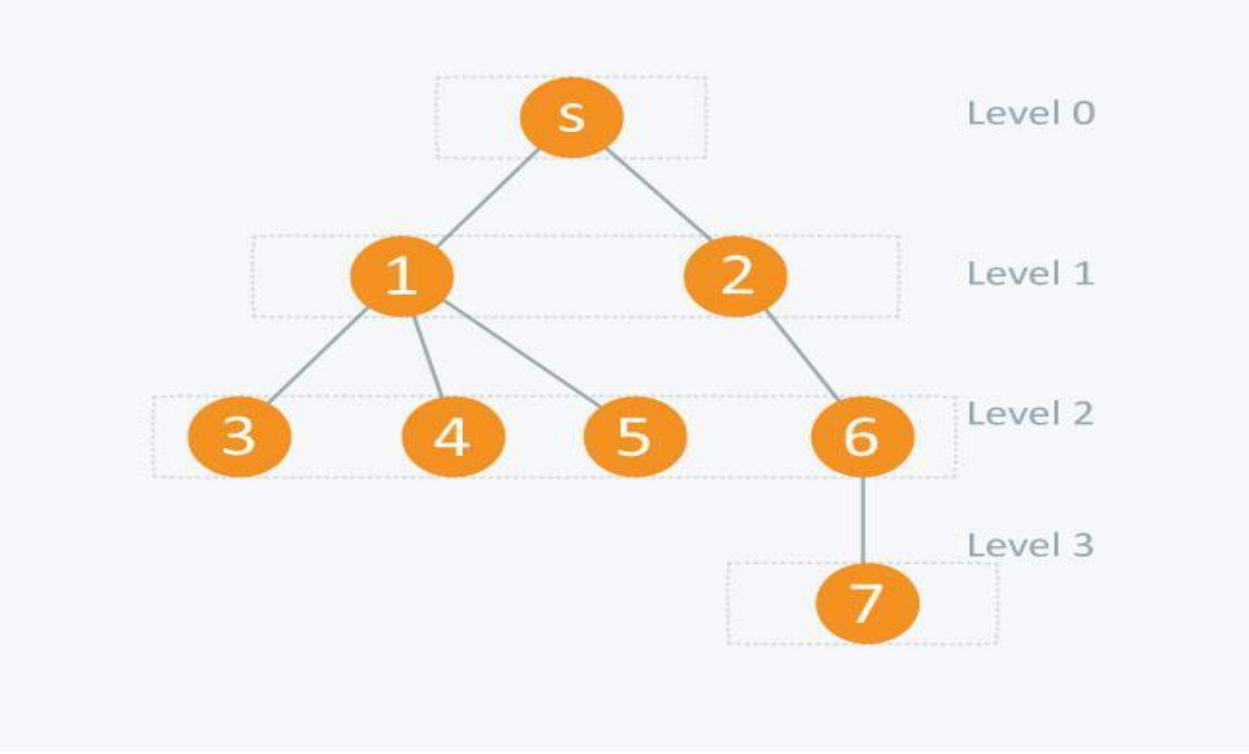


Example 1 :	
Input	Adjacency list representation of graph shown above. Source Vertex
Output	List of visited vertices

Exercise # 4: Level of each node in the graph

How to determine the level of each node in the given graph?

As you know, BFS involves a *level-order* traversal of a graph. Hence, you can use BFS to determine the level of each node as well.



Exercise # 4a: Write a function nodes_of_level(G, level) that takes a graph and level number as input and returns a list of all the nodes that are on the given level.

Example 1 :	
Input	Adjacency list representation of graph shown above. level = 1
Output	[1, 2]

Example 2 :	
Input	Adjacency list representation of graph shown above. level = 2
Output	[3, 4, 5, 6]

Exercise # 4b:

Write a function get_node_level(G, node) that takes a graph and node as input and returns its level.

Example 1 :	
Input	Adjacency list representation of graph shown above. node = 2
Output	1

Example 2 :	
Input	Adjacency list representation of graph shown above. node = 6
Output	2