#35: Graph Traversals

April 16, 2018 · *Wade Fagen-Ulmschneider*

Running Times of Classical Graph Implementations

	Edge List	Adj. Matrix	Adj. List
Space	n+m	n^2	n+m
insertVertex	1	n	1
removeVertex	m	n	deg(v)
insertEdge	1	1	1
removeEdge	1	1	1
incidentEdges	m	n	deg(v)
areAdjacent	m	1	min(deg(v), deg(w))

Implementations and Use Cases

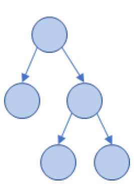
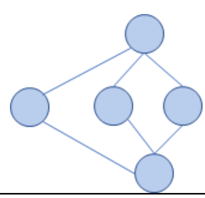
Ex. 1:

Ex. 2:

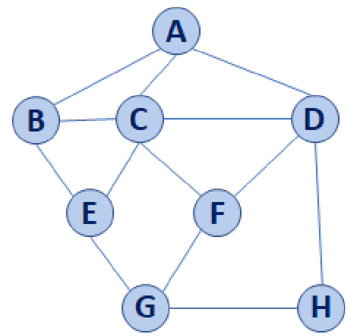
Graph Traversal

Objective: Visit every vertex and every edge in the graph.
Purpose: Search for interesting sub-structures in the graph.

We've seen traversal before – this is different:

BST	Graph
	

BFS Graph Traversal



A	
B	
C	
D	
E	
F	
G	
H	

Pseudocode for BFS

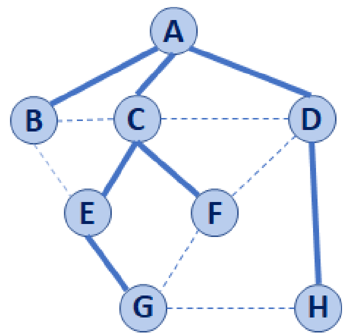
```
BFS(G):
  Input: Graph, G
  Output: A labeling of the edges on
      G as discovery and cross edges

  foreach (Vertex v : G.vertices()):
    setLabel(v, UNEXPLORED)
  foreach (Edge e : G.edges()):
    setLabel(e, UNEXPLORED)
  foreach (Vertex v : G.vertices()):
    if getLabel(v) == UNEXPLORED:
      BFS(G, v)

BFS(G, v):
  Queue q
  setLabel(v, VISITED)
  q.enqueue(v)

  while !q.empty():
    v = q.dequeue()
    foreach (Vertex w : G.adjacent(v)):
      if getLabel(w) == UNEXPLORED:
        setLabel(v, w, DISCOVERY)
        setLabel(w, VISITED)
        q.enqueue(w)
      elseif getLabel(v, w) == UNEXPLORED:
        setLabel(v, w, CROSS)
```

BST Graph Observations



1. Does our implementation handle disjoint graphs? How?

    a. How can we modify our code to count components?

2. Can our implementation detect a cycle? How?

    a. How can we modify our code to store update a private member variable `cycleDetected_`?

3. What is the running time of our algorithm?

4. What is the shortest path between **A** and **H**?

5. What is the shortest path between **E** and **H**?

    a. What does that tell us about BFS?

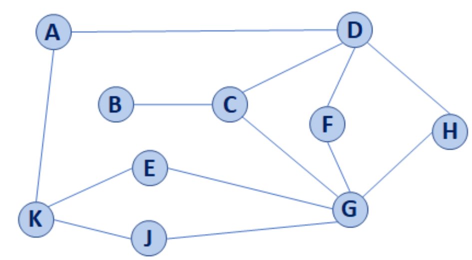
6. What does a cross edge tell us about its endpoints?

7. What structure is made from discovery edges in **G**?

<u>Big Ideas: Utility of a BFS Traversal</u>

Obs. 1: Traversals can be used to count components.
Obs. 2: Traversals can be used to detect cycles.
Obs. 3: In BFS, **d** provides the shortest distance to every vertex.
Obs. 4: In BFS, the endpoints of a cross edge never differ in distance, d, by more than 1: **|d(u) - d(v)| = 1**

Depth First Search – A Modification to BFS



Two types of edges: 1.

2.

<u>Running Time of DFS:</u>

Labeling:
- Vertex:
- Edge:

Queries:
- Vertex:
- Edge:

CS 225 – Things To Be Doing:

1. Programming Exam C starts Tuesday 4/17
2. MP6 due tonight, Monday, April 16th
3. lab_graphs available Wednesday
4. Daily POTDs are ongoing!