BFS Analysis

Learning Objectives

1. Know what BFS can be used for
2. Know the runtime of BFS

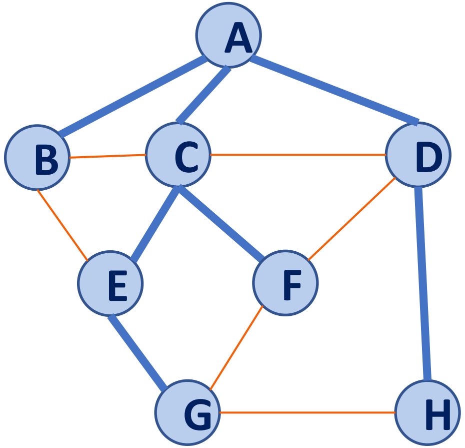
BFS Observations

Q: What is a shortest path from **A** to **H**?

Q: What is a shortest path from **E** to **H**?

Q: What structure is made from discovery edges?

Q: How does a cross edge relate to **d**?



v	d	P	Adjacent Edges
A	0	-	C B D
B	1	A	A C E
C	1	A	B A D E F
D	1	A	A C F H
E	2	C	B C G
F	2	C	C D G
G	3	E	E F H
H	2	D	D G

BFS Observations

Obs. 1: BFS can be used to count components.

Obs. 2: BFS 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:

$|d(u) - d(v)| = 1$

Graph Traversal - 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)
```

Discovery Edge

- An edge taken to find a new node

Cross Edge

- An edge taken to find an already visited node

Graph Traversal - BFS from a vertex

```
BFS(G, v):
  Queue q
  q.enqueue(v)

  while !q.empty():
    v = q.dequeue()
    If (getLabel(v) != VISITED){
        setLabel(v, VISITED)
        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)
    }
```

Node Labels

- Visited
- Unexplored

Edge Labels

- Discovery
- Cross
- Unexplored