CS 367 - Introduction to Data Structures
Tuesday, November 29, 2016

Program 5 assigned Monday?

Homework h9 available tomorrow

Last Time

ADTs/Data Structures

Graphs

- terminology
- implementation
- edge representations (A-M and AL)

Today

Graphs

- Traversal Algorithms
  - DFS
  - BFS
- Topological ordering
- Applications of BFS/DFS

Next Time

Read: finish *Graphs*, start *Hashing*

Graphs

- more terminology
- Dijkstra's algorithm

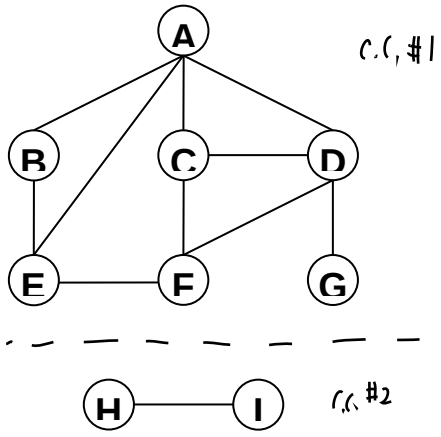
Hashing

- terminology

Searches and Traversals

Search Look through collection until found
- must pick a start vertex
  - if not, CS 367 convention

Traversal - visit each vertex exactly once.
need a start vertex - and visit vertices that are reachable



→ Which connected component in the graph above can produce the longest path?
#1

✵ CS367 Convention: Pick next unvisited vertex in increasing numerical or alphabetical order

Depth-First Search (DFS)

- Assume all vertices unvisited
- Relies on a stack.
  call stack: recursion

Algorithm

```
DFS(v)
    mark v as visited.
    for each unvisited successor(s) adjacent
        DFS(s)
```

* equiv to pre-order

Breadth-First Search (BFS)

- Assume all vertices unvisited at start.
- Relies on queue
  ↖ FIFO

Algorithm

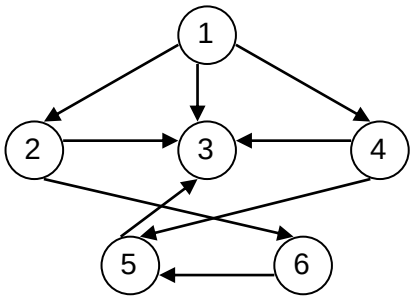
```
BFS(v)
  q=new Queue();
  mark v as visited;
  q.enqueue(v)  // priming the pump
  while q is not empty;
      c. dequeue();
      for each unvisited successor(s)
          mark s as visited
          q.enqueue(s);
```

✵ Equiv to level order traversal

Topological Ordering

IDEA:

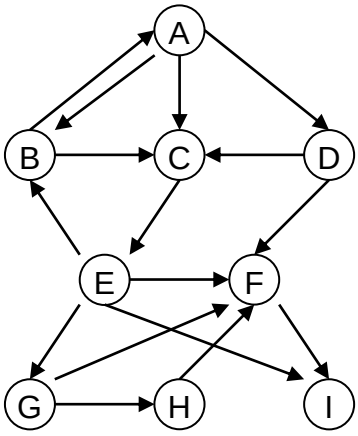
Example



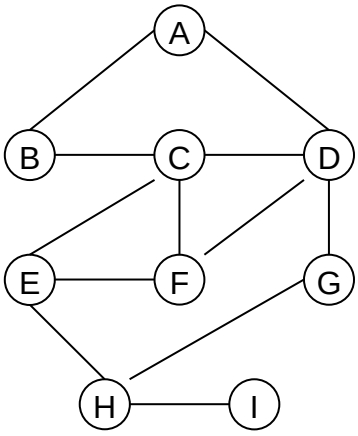
Iterative Algorithm (see readings for recursive algorithm)

DFS Practice

Graph 1



Graph 2



→ Give the order that vertexes are visited for depth-first search (DFS) starting at A.

Graph 1:

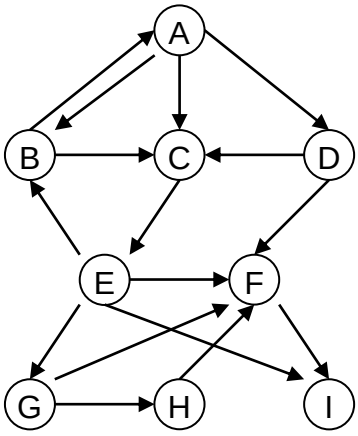
Graph 2:

→ Give the DFS spanning tree starting at A.

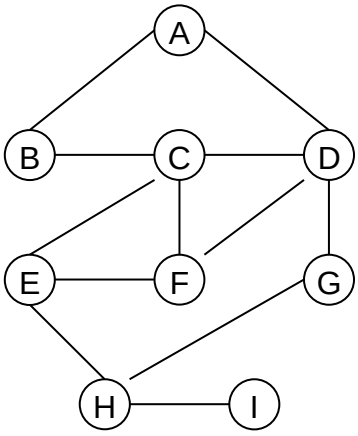
Graph 1: Graph 2:

BFS Practice

Graph 1



Graph 2



→ Give the order that vertexes are visited for breadth-first search (BFS) starting at A.

Graph 1:

Graph 2:

Give the BFS spanning tree starting at A.

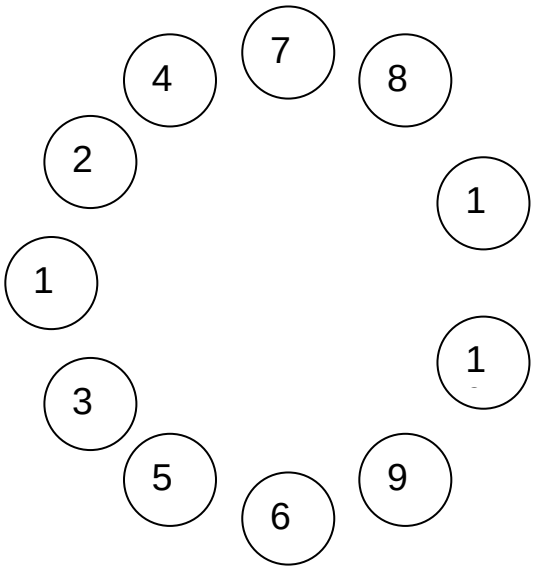
Graph 1: Graph 2:

Topological Ordering Practice

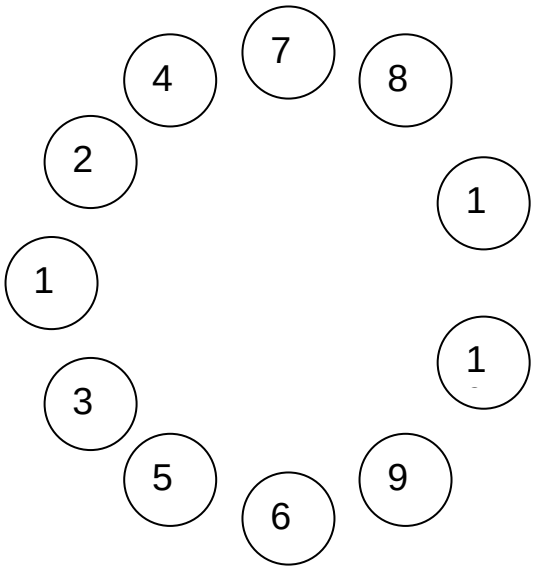
1. **Draw the edges indicating which steps must come before which other steps.**
2. **Simplify the graph.**
3. **Find set (unique) of possible topological orderings.**

1. get bread
2. get jelly
3. get peanut butter
4. get butter knife
5. open jelly
6. open peanut butter
7. take bread slice 1
8. take bread slice 2
9. use knife to spread jelly on bread slice
10. use knife to spread peanut butter on bread slice
11. put slices together with spreaded sides facing each other

Full Graph



Simplified Graph



Topological Orderings

Applications of DFS/BFS

Path Detection

Cycle Detection

More Graph Terminology