Lesson: Depth and Breadth First Search Algorithms

Goals

- Breadth First Search
- Depth First Search

1 COVID-19 Contact Tracing Example

Suppose that there is an outbreak of COVID-19 within a small group of the 1^{st} company. We want to conduct a contact tracing study to see who needs to quarantine. Consider the following details.

- We know that Professor Curry contracted COVID-19. (I don't have COVID-19. This is just an example.)
- We want to study a small group of ten possible mids that *could* because they are supposed to be traveling on an MO tomorrow. If they are found in the contact tracing tree, then they are unable to go on the MO. Set of students {Jim, Dre, Sean, Connor, Ana, Caroline, Mike, Jayla, Xavier, Ashley}.
- We know that Jayla, Xavier, and Caroline were all in Prof Curry's office without masks on.
- Jayla, Caroline, and Ana are all roommates.
- Xavier and Mike are roommates.
- Mike, Xavier, and Connor all ate lunch inside together today.

1. Given what you know now about algorithms, formulate an algorithm for solving this problem.

2 Family Tree Example

Let's take a look at another example. We want to come up with an algorithm for creating a visual representation of the Simpson's family tree. Here's the relevant information:

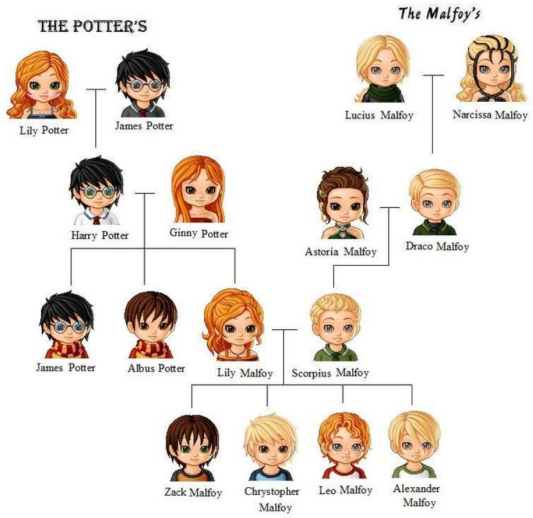
- Abraham and Mona are Herb and Homer's parents.
- Clancy and Jackie are Marge, Patty, and Selma's parents.
- Homer and Marge are Bart, Lisa, and Maggie's parents.
- Selma is Ling's Parent.

1. Create a visual representation of the Simpson's family tree.

2. Number each family member in the order in which you drew them.

3 Now, let's work backwards...

Consider the visual representation of (a portion of) the Potter and Malfoy family tree. (Image credit from Reddit)



1. Systemically determine the parent of each character depicted in the tree above.

4 Search Algorithms

Given s as the source node, the two search algorithms we study in this lession will systematically explore the edges of G to *discover* every node that is reachable from s. They also calculate the fewest number of edges from s to each reachable node in G. While very similar, these two search algorithms differ in implementation and analysis.

4.1 Breadth First Search

You can think of the Bread First Search (BFS) algorithm that explores each *layer* of the network.

To simplify this algorithm, the BFS signifies each node as *explored* when found in the tree.

- All nodes start as unexplored and may later be explored.
- The BFS will keep up with a queue Q that keeps up with the list of nodes that should be explored next.

The BFS constructs a BF ***tree***, initially containing only its root node s. Whenever the search encounters an unexplored node v in the course of scanning the adjacency list of an already encountered node u, the node v and the edge (u, v) are added to the tree. Thus, we refer to the node u as the ***parent*** of v. Since a node is discovered at most once, then each node has at most one parent.

What do you think are the pros and cons of the BFS algorithm?

Questions?

Written Steps:

Initialization: Create queue $Q = \emptyset$. Add source node s to Q, and mark s as explored.

1. If Q is empty, then proceed to Step 3. Otherwise, remove node v from the bottom of Q. Proceed to Step 2.

2. For some adjacent node $w \in V$ of v, if w is not explored, then add w to the bottom/end of Q, mark w as explored, and set the parent of w to be v. Repeat Step 2.

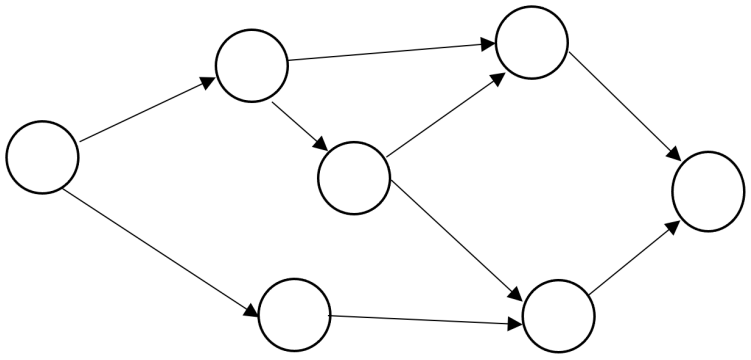
3. Terminate the algorithm with the BF tree resulting from the parent vector of each node in V.

Pseudocode:

Algorithm 1 Determine a BF tree for the Graph G with source node s

```
Let Q be the queue
Add s to Q.
Mark s as explored.
while Q is not empty do
  Remove v from Q
  for all adjacent nodes w ∈ 𝒱 of v  do
    if w is not explored then
      Add w to Q
      Mark w as explored
      Mark v as the parent of w
    end if
  end for
end while
```

Run the BFS algorithm on the following graph:



5 Depth First Search

Given the name of the Depth First Search (DFS), how do you think it differs from the BFS?

Just like the BFS algorithm, the DFS algorithm also constructs a ***tree***, initially containing only its root node s. Rather than exploring the *layers* of the graph (as with the BFS algorithm) by exploring **all** adjacent nodes of each node before moving to another node, the DFS instead continually explores a series of adjacent nodes until the algorithm reaches a node having no adjancent nodes (also known as a leaf node). The DFS algorithm is a repetitive algorithm that *backtracks* to a recently visited node.

What do you think are the pros and cons of the BFS algorithm?

Questions?

Written Steps:

Initialization: Create stack $S = \emptyset$. Add source node s to the top of S, and mark s as explored.

1. If S is empty, then proceed to Step 3. Otherwise, remove node v from the top of S. Proceed to Step 2.
2. For each adjacent node $w \in V$ of v, add w to the top of S and set the parent of w to be v. Otherwise, repeat Step 2.
3. Terminate the algorithm with the DF tree resulting from the parent vector of each node in V.

Pseudocode:

Algorithm 2 Determine a DF tree for the Graph G with source node s

```
Let S be a stack.
Insert s at the top of S.
while S is not empty do
  Remove node v from the top of S
  for all adjacent nodes w ∈ 𝒱 of v do
    Add w to the top S
    Mark v as the parent of w
  end for
end while
```

Run the DFS algorithm on the following graph: