Slides on Tutorial 4 Exercise

CPSC 457

Tutorial 4 (WEEK 2, Day 2)

TA: Michelle Nguyen

Please note:

- I will not post example solutions to the exercise on D2L
- Feel free to come to tutorials for help or if you want me to take a look at your solution

1. Please go to D2L → CPSC457→Tutorials→Michelle
2. Download the .zip called "Tutorial4Exercise".
3. Unzip the file

Please note:

- I will not post example solutions to the exercise on D2L
- Feel free to come to tutorials for help or if you want me to take a look at your solution

MOTIVATION

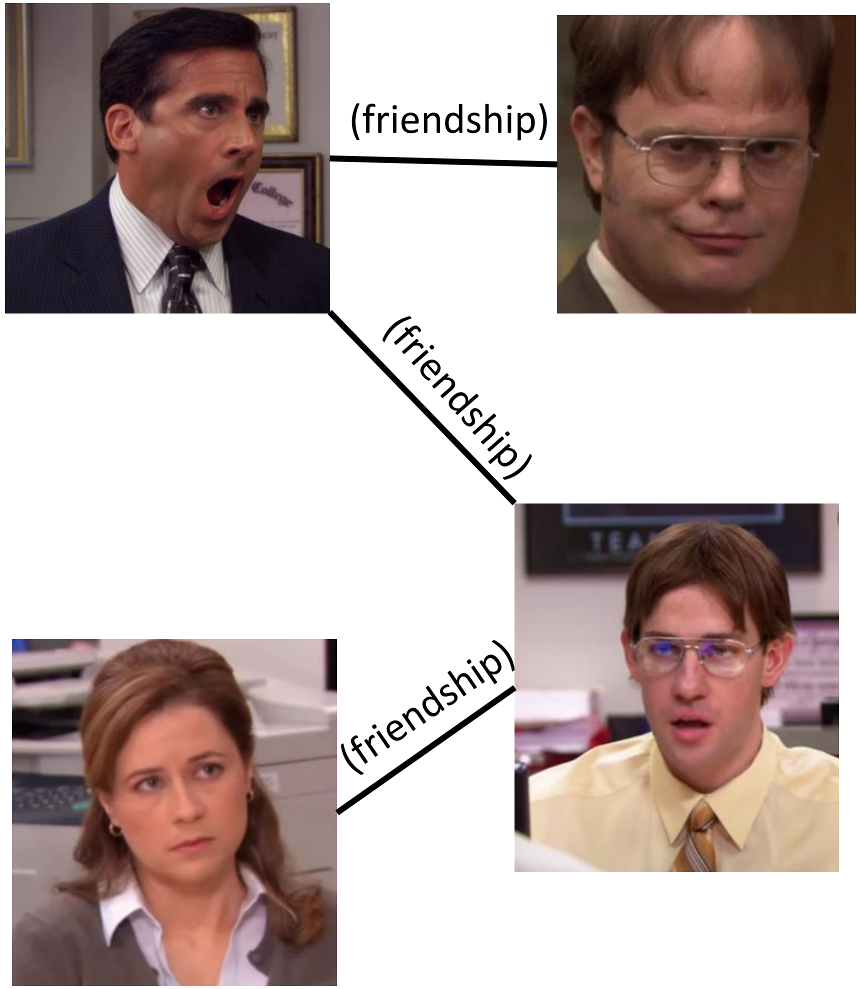
- Practice writing C/C++
  - structs, functions, strings, vectors, etc.
  - learning to use [new to you?] code with the help of a reference (cplusplus.com)
- Practice in the format of assignments (given some code, fill it in so it solves the problem)

Tutorial 4 Exercise: The Social NETWORK

With Covid and well, just about everything happening in 2020--- it's hard to know who you can trust. Perhaps if people are friends, or have mutual friends, or a series of friends between them then they can trust each other.

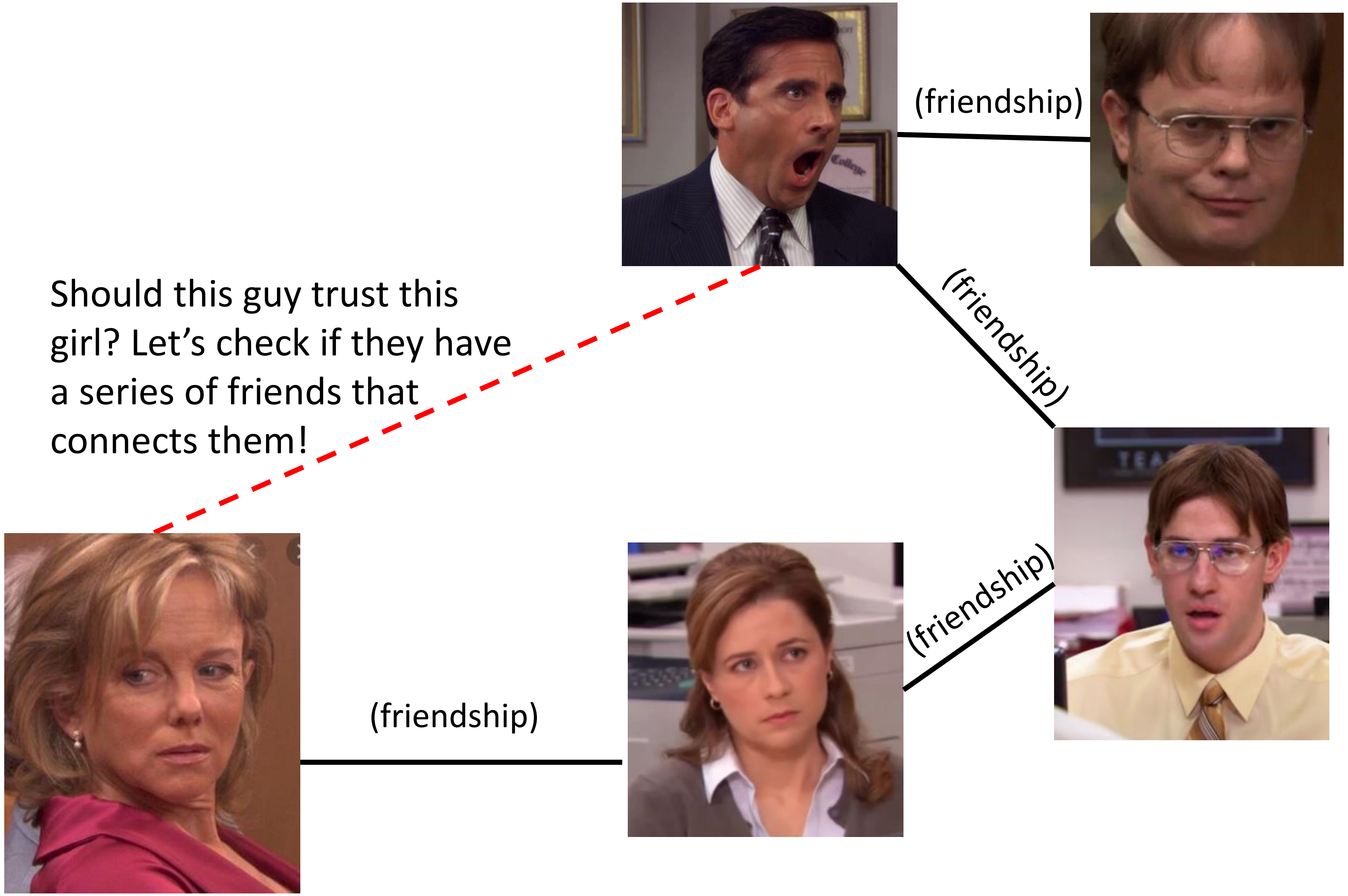
(Psst. Don't take this too seriously, it's just a tutorial exercise)

In this problem/tutorial exercise, you will develop a program that 'constructs' a social network from given input, and then when given two names of peoples, finds out whether they can trust each other (in other words, whether they are friends or have [a series] of friends between them).



Tutorial 4 Exercise: The Social NETWORK

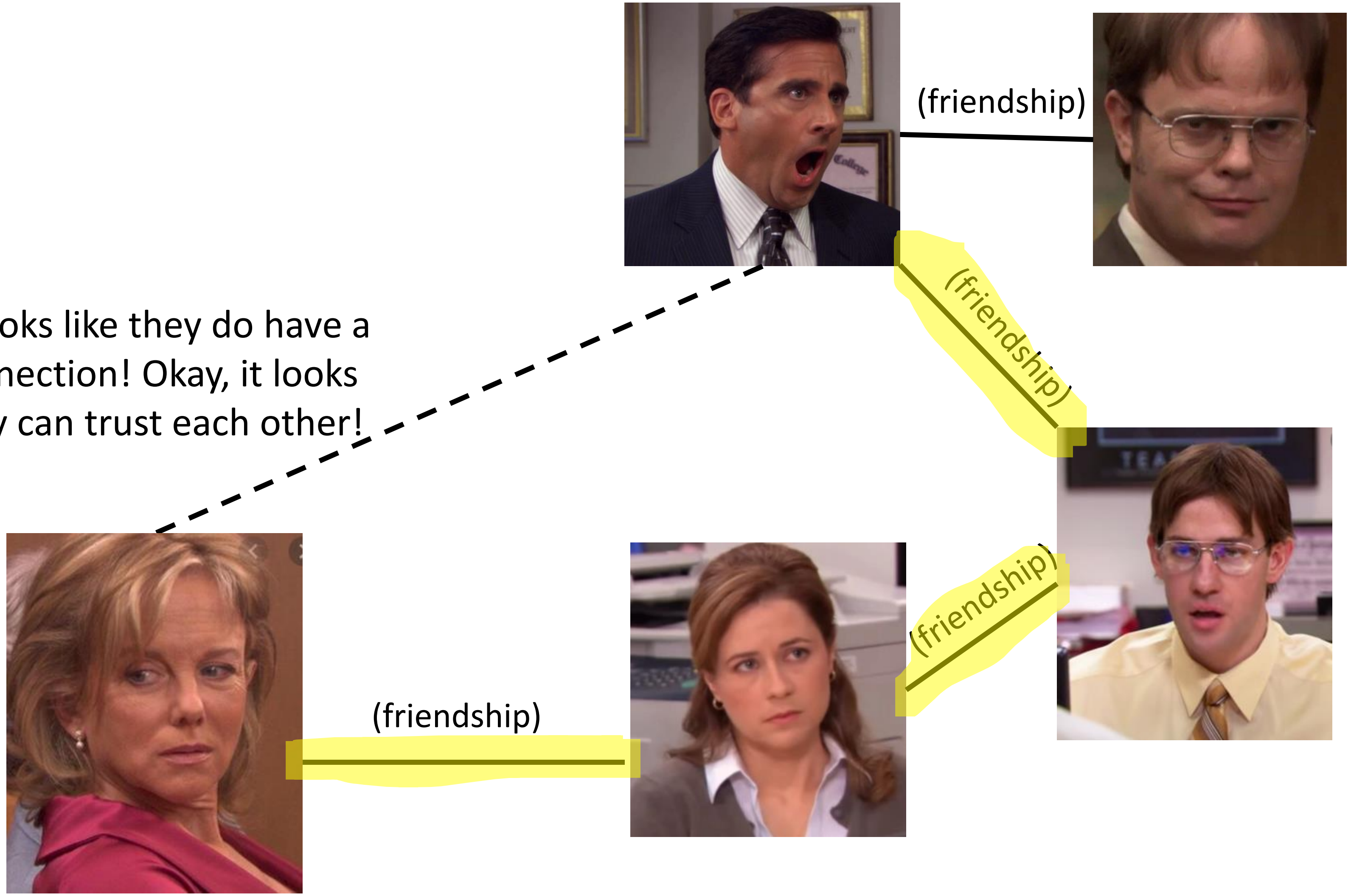
Example:



Should this guy trust this girl? Let's check if they have a series of friends that connects them!

Tutorial 4 Exercise: The Social NETWORK

Example:



It looks like they do have a connection! Okay, it looks they can trust each other!

Strategy

- Given a graph where people are nodes/vertices and friendships are edges
- Being asked if two people can trust each other (i.e. there is a path between them)

Strategy:

- Start with one of two the people asked about, traverse the social network graph using *depth first search* and see the people are connected by friendships

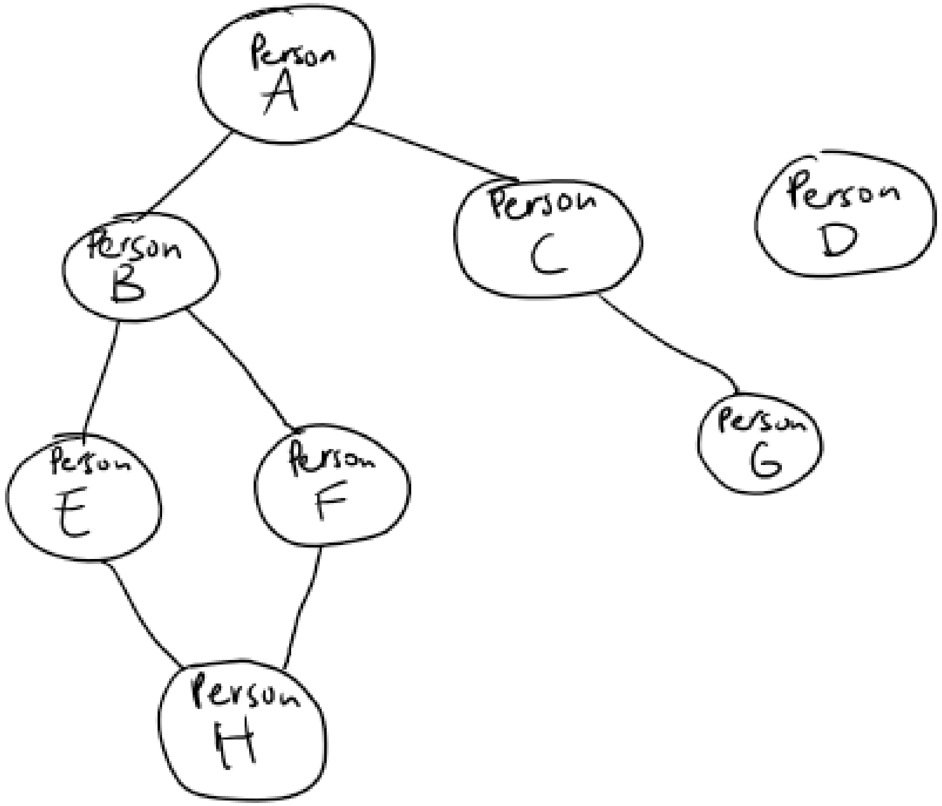
Reminder: DFS

- Should remember depth first search (dfs) from cpsc 331 or equivalent pre-req

- Quick re-cap demo

Adjacency
List:

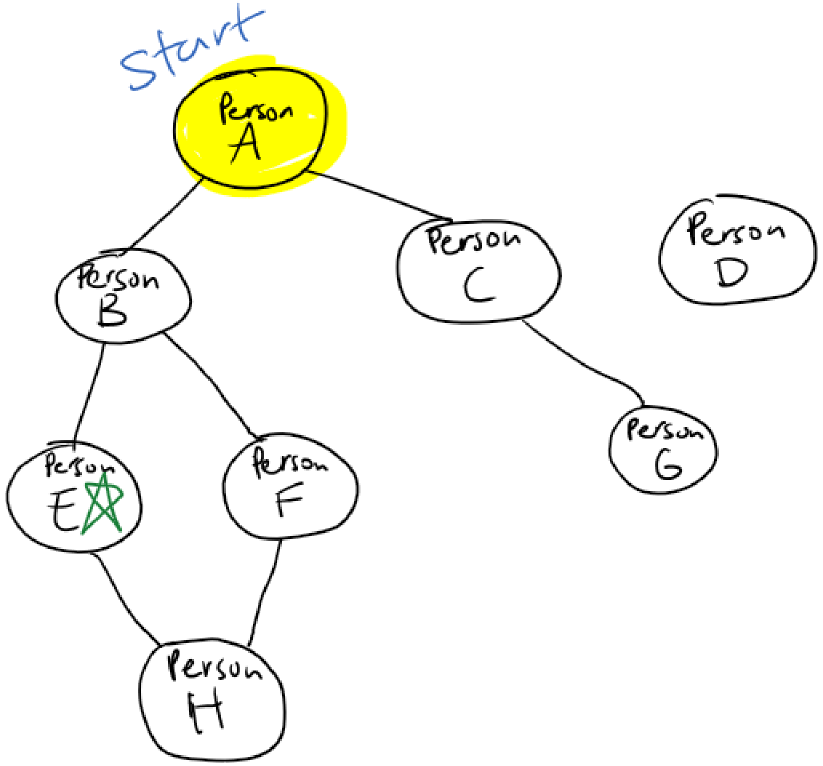
[
A : [B, C]
B : [E, F, A]
C : [G, A]
D : []
E : [B, H]
F : [B, H]
G : [C]
H : [E, F]
]



Problem: Is person A connected to person E?
Let's perform dfs to see!

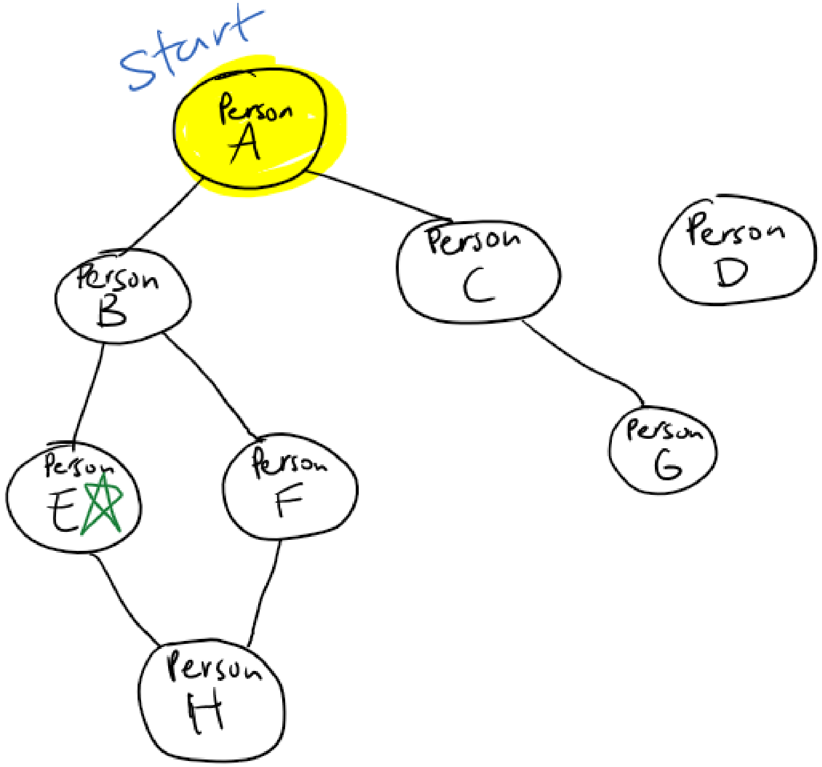
Nodes to check next

stack:



stack:

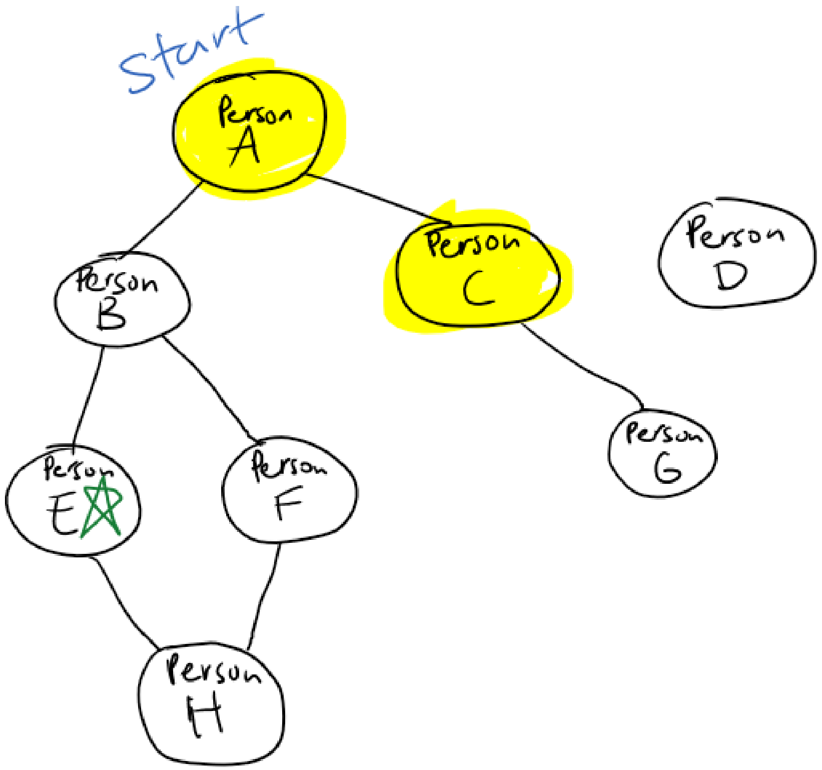
B, C


Start
Person A
Person B
Person C
Person D
Person E
Person F
Person G
Person H

stack:

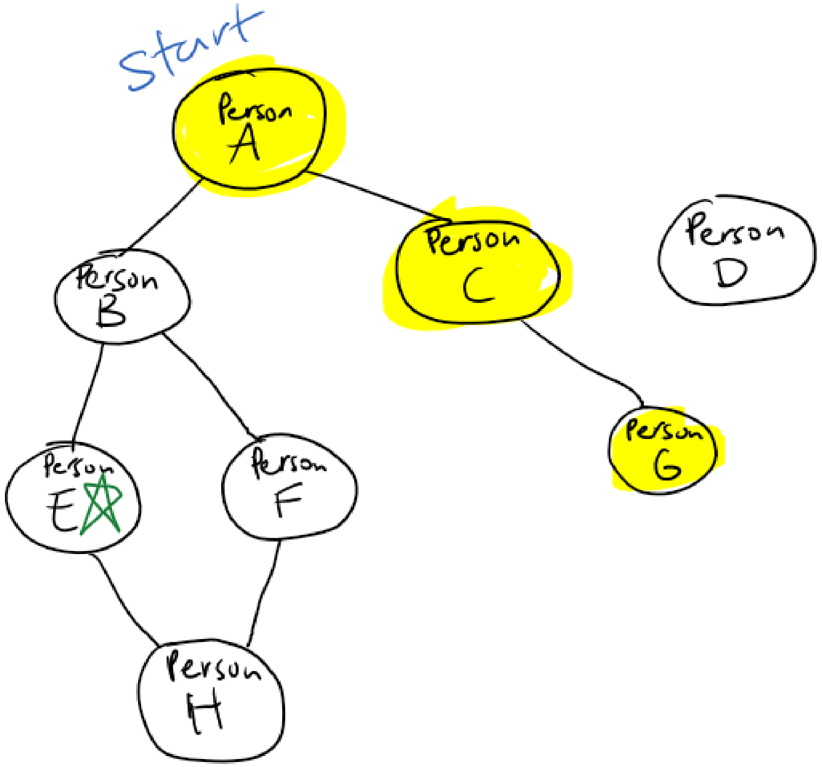
B, G, ~~A~~

↑ already visited A



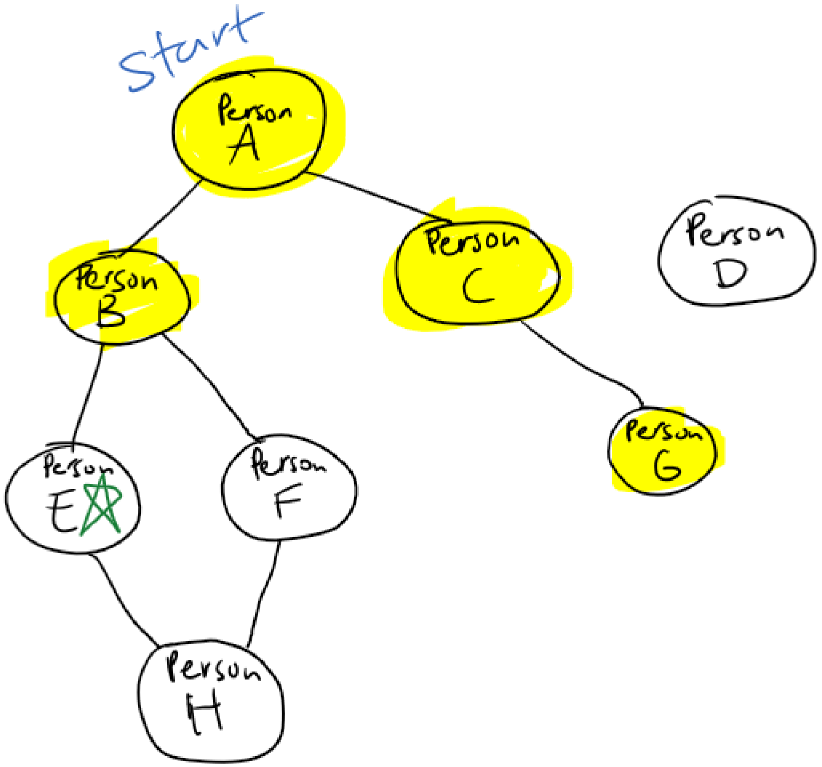
stack:

B


Start
Person A
Person B
Person C
Person D
Person E
Person F
Person G
Person H

stack:

E, F

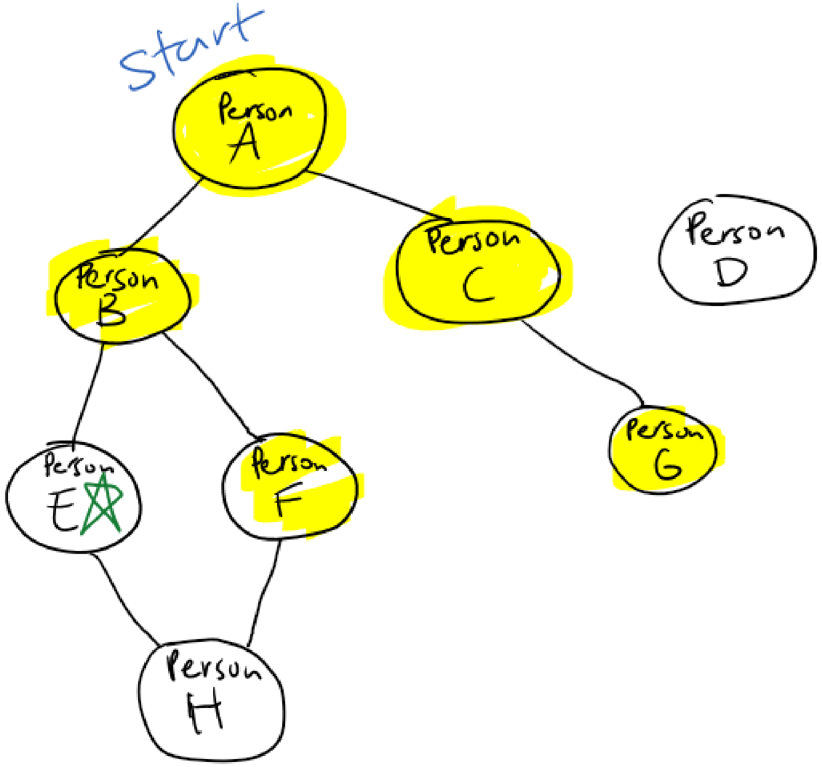


note: I skip a step of adding A to the stack and then popping it b/c it's trivial

stack:

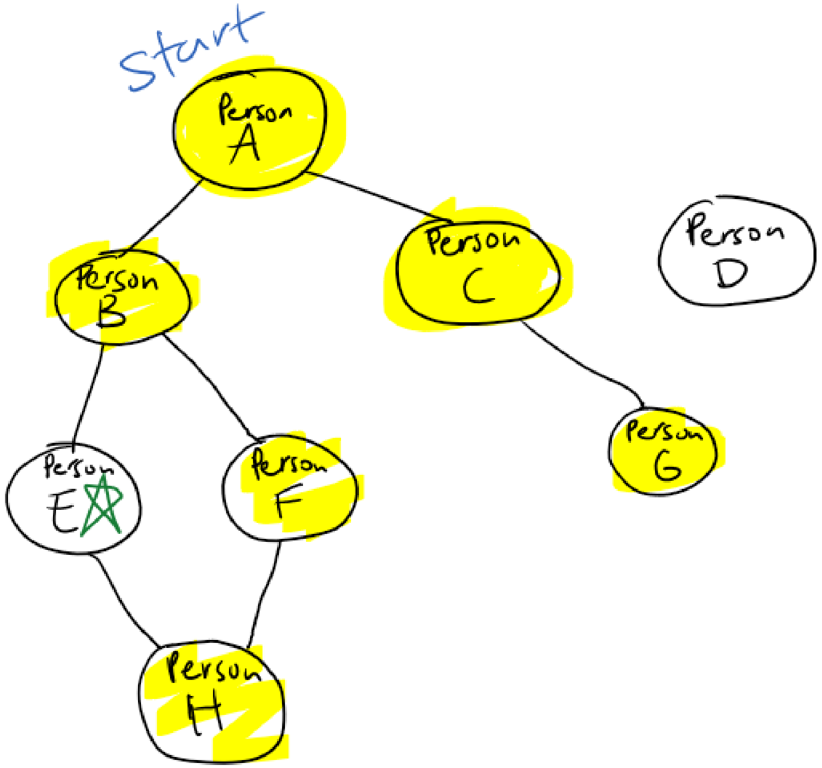
E, H, ~~B~~

^ already visited B



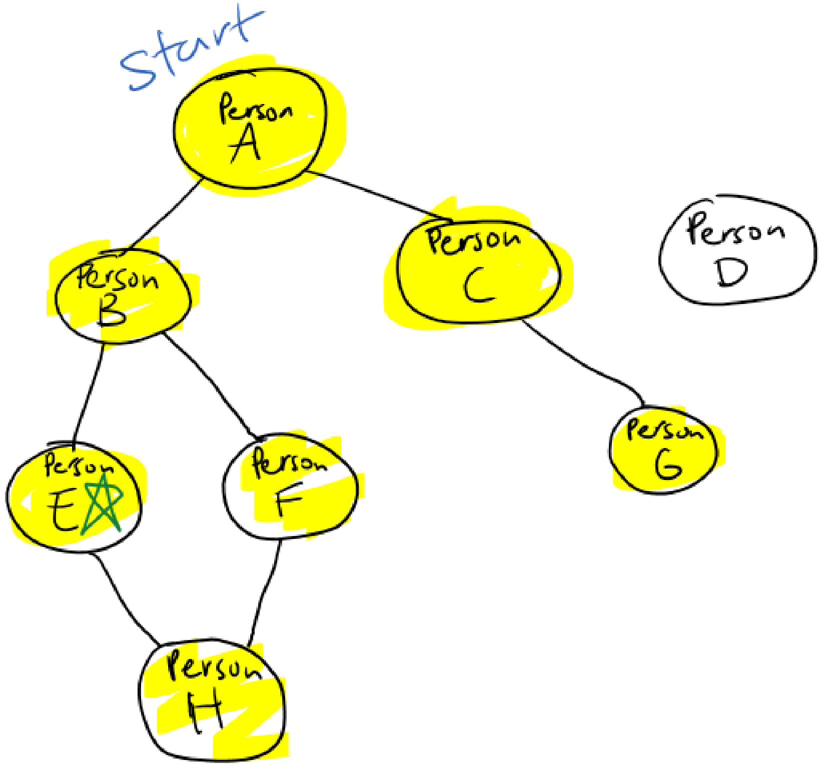
stack:

E , E


Start
Person A
Person B
Person C
Person D
Person E
Person F
Person G
Person H

stack:

E



Great! So E is connected to A!

Notice that if we were asked whether Person A is connected to Person D, we would have follow the procedure above (if we start with Person A) and then the stack would be empty without person D having ever appeared in the stack