

Confidential – DO NOT Copy

### MSI Pre-Interview Question: Shortest Path

#### Directions:

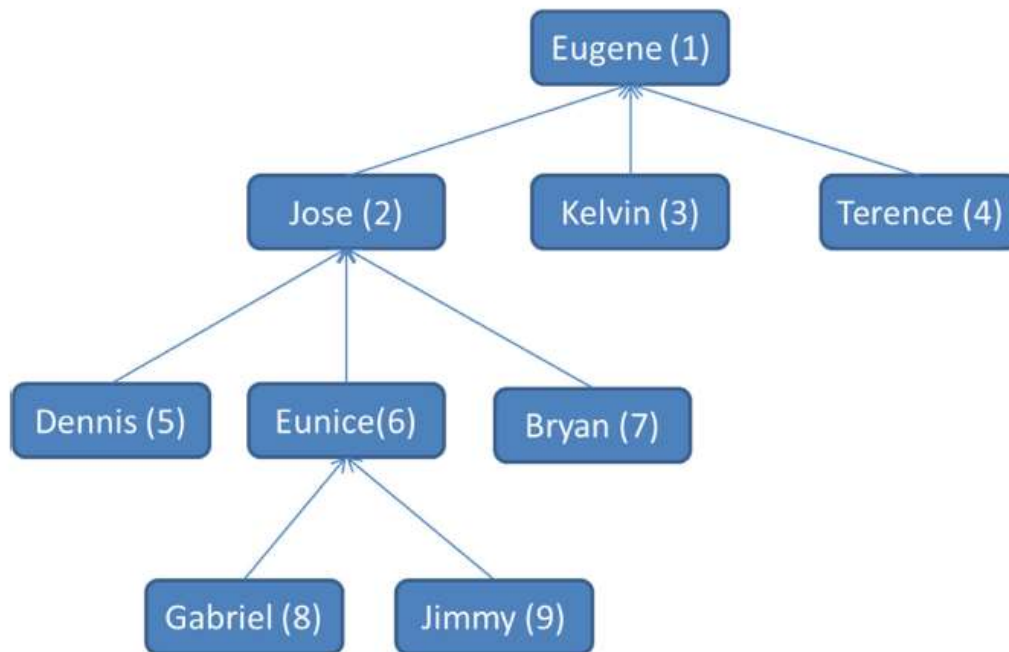
You'll have up to 60 minutes to complete the coding question and send it back to me, in a single text file. Here is some of the information you'll need for the code once it's sent, I wanted to give you the information ahead of time so you'll know what to expect.

Your task is to implement an interface we have defined, in any of the following languages: C++, Java, or C#. Your solutions must conform to these MANDATORY items:

1. **DO YOUR OWN WORK.** If it appears not to be your own work, it will be disqualified. In addition, limit your solution to the standard libraries available in your language. Do not use any databases, external libraries, or third party solutions.
2. **CODE MUST COMPILE.** Solutions must compile and function at a minimum for consideration however additional credit will be given for elegant and robust code.
3. **DOCUMENT ALL ASSUMPTIONS.** Explain why you made that assumption. You may make the wrong assumption, but if it's well-documented, you'll help us see your thought process. Undocumented assumptions can't help us help you.
4. **DESCRIBE THE RUNTIME AND SPATIAL COMPLEXITY OF YOUR SOLUTIONS.** This must be stated somewhere in your solution for each of the questions, preferably in the comments above each solution. Runtime and spatial analyses are core elements of software engineering. Because of this, solutions without runtime and spatial analyses will not be considered.
5. **PROVIDE THE MOST EFFICIENT SOLUTION POSSIBLE.** We will be evaluating the quality of your algorithm.

### The Question:

Infotech is a company which has CEO Eugene and a hierarchy of employees. Please implement the `shortestPath` method to find the shortest path between two employees.



### Sample Data

CEO Eugene has 3 employees reporting to him: {Jose, Kelvin, Terence}

Jose has three reports {Dennis, Eunice, Bryan}

Kelvin has no reports {}

Terence has no reports {}

Dennis has no reports {}

Eunice has 2 reports {Gabriel, Jimmy}

Bryan has no reports {}

Gabriel has no reports {}

Jimmy has no reports {}

### Sample calls

`shortestPath (Eugene, Gabriel, Jimmy) = Gabriel > Eunice > Jimmy`

`shortestPath (Eugene, Jimmy, Bryan) = Jimmy > Eunice > Jose > Bryan`

`shortestPath (Eugene, Jimmy, Kelvin) = Jimmy > Eunice > Jose > Eugene > Kelvin`

`shortestPath (Eugene, Eunice, Jimmy) = Eunice > Jimmy`