

Unimarket Java Pre-Interview Test

This should take about 30 minutes for each section. We are not expecting very long answers. One page for each section is plenty.

Java Coding

Given a Directed Acyclic Graph (i.e. it has no loops) that looks like `GraphNode` below, write an algorithm to determine the length of the shortest path between two nodes. Length is defined as the number of steps required to get from `startNode` to `finishNode`.

The answer must be written in Java. Do not alter the `GraphNode` interface. Your implementation of `getShortestPathDistance` should not know what class implements `GraphNode` but you can assume `equals()` and `hashCode()` work correctly on `GraphNode`. You can create and use additional helper classes and/or methods, just don't alter `GraphNode`.

We will be running your code to test for correctness. We recommend you compile and run it before submission. It will judged on both correctness and code quality.

```
public interface GraphNode {
    /**
     * Returns all the GraphNodes directly linked
     * to this GraphNode.
     * These are considered to be distance 1 from this node.
     */
    Set<GraphNode> getDirectlyLinkedNodes();
}

public static int getShortestPathDistance(
    GraphNode startNode,
    GraphNode finishNode) {
    // implement this
}
```

Did you use a depth-first or breadth-first search? Why?

If you were to write unit tests for this method, what test cases would you include?

What is the performance of your algorithm in Big O notation? Please explain the meaning of any variables used in your Big O answer.

Design

Imagine you are implementing the game chess. What are the key classes you would use to represent the state of the game, and what would their responsibilities be? You can describe this any way you wish - text, UML, code snippets, pseudo-code, etc. This is not a test of your UML or other diagramming abilities; just ensure your intention is clear and the design is good. Consider the following problems

- How would you represent the different types of pieces?
- How would you model which pieces are where?
- How would you determine where the available moves are for each piece?
- How would you determine which pieces can be captured?
- How would you determine if the current player is in check?