

## Problem 2

When coming up with test cases, I focused on testing for the expected result, possible representation exposure, any edge cases I could think of, and inputs that should throw exceptions. For the expected result, I just used three different lists of nodes and edges (one empty, one regular, and one with duplicates). If I wanted to make it even more extensive for any additional checks, I could've made more lists with more/less nodes and edges, but I thought what I made was fine enough for this. For possible exposure, I tried modifying the output of methods that returned anything that could possibly affect the graph's underlying representation. For edge cases and throws, I focused on what I wrote about in my specs to see if the results were expected when doing something (ex: give a node not in the graph to `outgoingEdges()`, give a null node to `outgoingEdges()`, give a list of duplicate nodes to graph constructor, etc).