Homework 4: Problem 2 - Testing Strategy

1 Testing Strategy

The testing strategy follows several key principles to ensure comprehensive validation of the Graph ADT implementation:

1. **Black-Box Testing**: Since we're performing black-box testing, the tests focus on validating behavior based solely on the API specification without knowledge of the internal implementation.
2. **Incremental Complexity**: Tests start with the simplest functionality (empty graph, single node) and progressively test more complex graph structures and operations.
3. **Boundary Cases**: test boundary conditions such as empty graphs, single nodes, reflexive edges, and multiple edges between the same nodes.
4. **Edge Cases**: test various edge cases including reflexive edges, multiple edges between the same nodes with the same or different labels, and the ordering requirements for node and edge listing.
5. **Examples from Assignment**: include specific tests for the examples provided in the assignment description to ensure the implementation meets the exact requirements.
6. **0-1-2 Heuristic**: As suggested in the assignment, apply the "0, 1, 2" case analysis:
   - Testing with 0, 1, and 2 nodes
   - Testing with 0, 1, and 2 edges between nodes
   - Testing nodes with 0, 1, and 2 children
   - Testing nodes with 0, 1, and 2 parents

The test suite for the GraphWrapper class (and by extension, the Graph ADT) includes the following key test categories:

1.1 Basic Graph Structure Tests

- **Empty Graph**: Verify a newly created graph contains no nodes or edges
- **Single Node**: Test adding a single node and verify it exists
- **Multiple Nodes**: Test adding multiple nodes and verify the correct lexicographical ordering

1.2 Edge Management Tests

- **Single Edge**: Test adding a single edge and verify it exists
- **Multiple Edges**: Test adding multiple edges between different nodes
- **Multiple Edges Between Same Nodes**: Test adding multiple edges between the same nodes
- **Reflexive Edges**: Test adding edges from a node to itself

1.3 Special Case Tests

- **Figure 3 Example**: Test the specific example from Figure 3 in the assignment
- **ListChildren Example**: Test the specific example for listChildren("a")
- **ListChildrenXML Example**: Test the specific example for listChildrenXML("a")

1.4 Ordering Requirements Tests

- Test that nodes are listed in lexicographical order
- Test that child nodes with edges are listed in lexicographical order by node name
- Test that edges with the same nodes are secondarily ordered by edge label