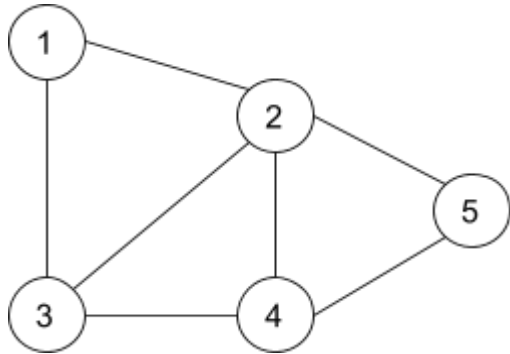
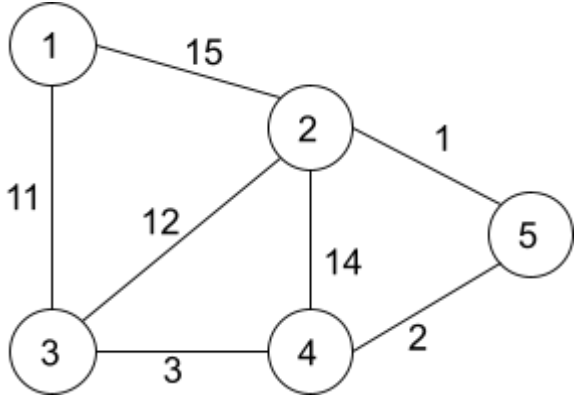


Unit test design


Configuration of the scenes

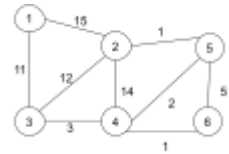
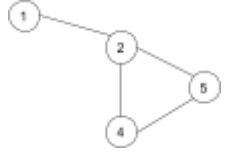
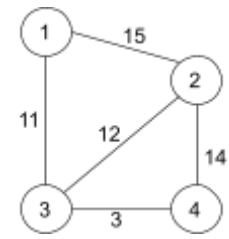
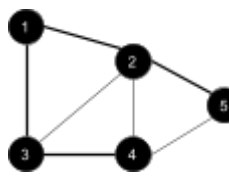
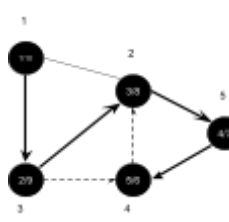
Name	Class	Scenes
graphScenary1	GraphTest	
graphScenary2	GraphTest	

Name	Class	Scenes
listScenary1	AdjListTest	
listScenary2	AdjListTest	

Design of test cases for each data structure

Graph methods

Purpose of the test: Verify that the methods in the graph works correctly.				
Class	Method	Scenes	Inputs	Results
Graph	testInsert	graphScenary1	Vertex six = new Vertex(6) List<Vertex<T>> adjacent = {2, 5}	

Graph	testInsert2	graphScenary2	Vertex seventh = new Vertex(6) List<Vertex<T>> adjacent = {4, 5} List<Integer> weights = {1, 5}	
Graph	testRemove	graphScenary1	Vertex three = new Vertex(3)	
Graph	testRemove	graphScenary2	Vertex five = new Vertex(5)	
Graph	testBfs	graphScenary1	Vertex origin = new Vertex(1)	
Graph	testDfs	graphScenary1	Vertex origin = new Vertex(1)	
Graph	testDijkstra	graphScenary2		Finds the minimum path between vertex 1 and the other vertices.
Graph	testFloyd	graphScenary2		Finds the minimum path between each pair of vertices.
Graph	testPrim	graphScenary2		Returns a total of 23
Graph	testKruskal	graphScenary2		Finds a minimum spanning tree.

Purpose of the test: Verify that the methods in the adjacency list works correctly.

Class	Method	Scenes	Inputs	Results
AdjList	testInsertVertex	listScenary1		
AdjList	testInsertVertex	listScenary2		
AdjList	testInsertEdge	listScenary1		
AdjList	testInsertEdge	listScenary2		
AdjList	testBfs	listScenary1		
AdjList	testBfs	listScenary2		
AdjList	testDfs	listScenary1		
AdjList	testDfs	listScenary2		
AdjList	testDijkstra	listScenary1		
AdjList	testDijkstra	listScenary2		
AdjList	testFloyd	listScenary1		
AdjList	testFloyd	listScenary2		
AdjList	testPrim	listScenary1		
AdjList	testPrim	listScenary2		
AdjList	testKruskal	listScenary1		
AdjList	testKruskal	listScenary2		