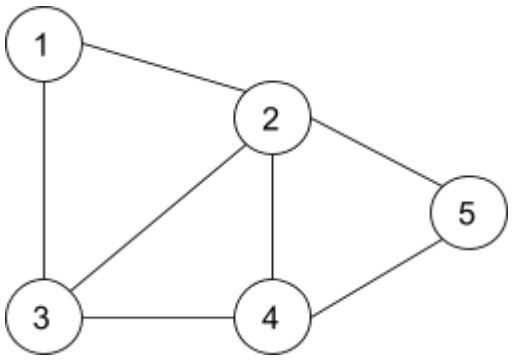
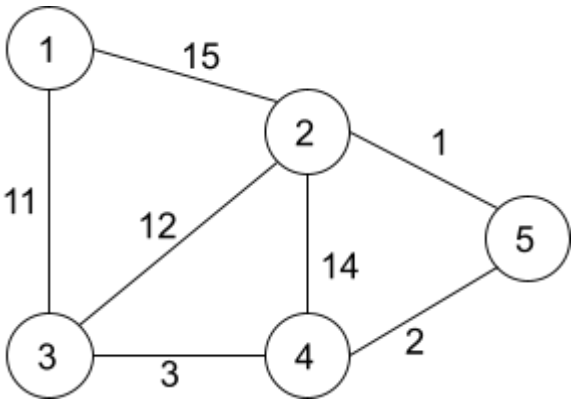
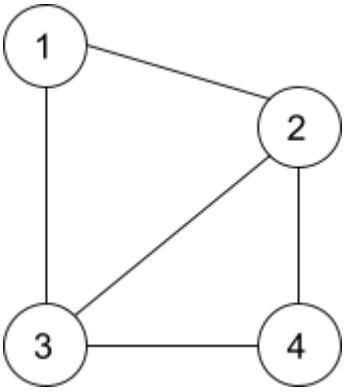
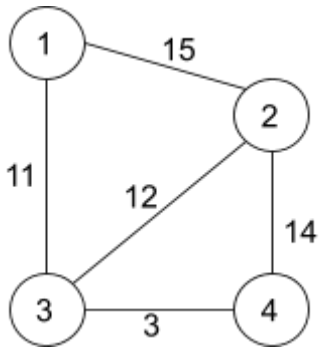
Unit test design
Configuration of the scenes

Name	Class	Scenes
graphScenary1	GraphTest	
graphScenary2	GraphTest	

Name	Class	Scenes
graphScenary1	MatrixGraphTest	

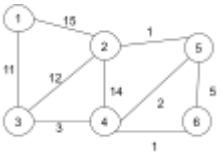
graphScenary2	MatrixGraphTest	

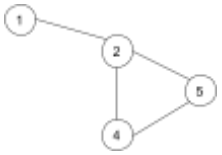
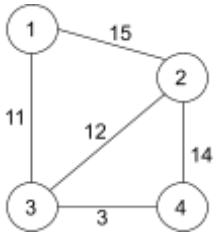
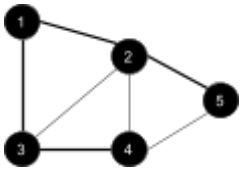
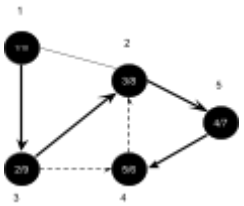
Name	Class	Scenes

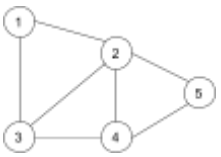
graphScenary1	MountainTest	int [] f = {5,2};
		
		int [] lm = {1,2,2};
		int [] lm2 = {2,4,2};
		int [] lm3 = {1,3,3};
		int [] lm4 = {3,6,3};
		int [] lm5 = {3,5,1};
		
		m.insertLandMarks(5);
		m.insertLandMarks(lm);
		m.insertLandMarks(lm2);
		m.insertLandMarks(lm3);
		m.insertLandMarks(lm4);
		m.insertLandMarks(lm5);
		
		m.addFriends(f);

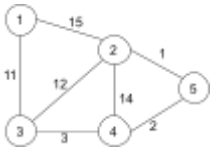
Design of test cases for each data structure

Graph methods

Purpose of the test: Verify that the methods in the graph works correctly.				
Class	**Method**	**Scenes**	**Inputs**	**Results**
ListGraph	testInsert	graphScenary1	Vertex six = new Vertex(6) List<Vertex<T>> adjacent = {2, 5}	
ListGraph	testInsert2	graphScenary2	Vertex seventh = new Vertex(6) List<Vertex<T>> adjacent = {4, 5}	

			List<Integer> weights = {1, 5}	
ListGraph	testRemove	graphScenary1	Vertex three = new Vertex(3)	
ListGraph	testRemove	graphScenary2	Vertex five = new Vertex(5)	
ListGraph	testBfs	graphScenary1	Vertex origin = new Vertex(1)	
ListGraph	testDfs	graphScenary1	Vertex origin = new Vertex(1)	
ListGraph	testDijkstra	graphScenary2		Finds the minimum path between vertex 1 and the other vertices.
ListGraph	testFloyd	graphScenary2		Finds the minimum path between each pair of vertices.
ListGraph	testPrim	graphScenary2		Finds a minimum spanning tree.
ListGraph	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**
MatrixGraph	testInsertVertex	graphScenary1	Vertex five = new Vertex(5) List<Vertex<T>>	

			adjacent = {2, 4}	
MatrixGraph	testInsertVertex	graphScenary2	Vertex five = new Vertex(5) List<Vertex<T>> adjacent = {2, 4} List<Integer> weights = {1, 2}	
MatrixGraph	testDijkstra	graphScenary2		Finds the minimum path between vertex 1 and the other vertices.
MatrixGraph	testFloyd	graphScenary2		Finds the minimum path between each pair of vertices.
MatrixGraph	testPrim	graphScenary2		Finds a minimum spanning tree.
MatrixGraph	testKruskal	graphScenary2		Finds a minimum spanning tree.

Model Methods

Purpose of the test: Verify that the methods in the adjacency list works correctly.				
Class	**Method**	**Scenes**	**Inputs**	**Results**
Mountain	testInsertLandMarks	setupScenary1	landmarks = new ListGraph<>() A = 2 B = 4 C = 10	The landmark is added correctly.
Mountain	testAddFriends	setupScenary1	A specific landmark.	The friend is added correctly.
Mountain	testCalcMinEnergy	setupScenary1		Shows the minimum energy needed to visit all the friends in the mountain.