UNIVERSIDAD ICESI

Algoritmos y Programación II
Facultad de ingeniería

Formato de escenarios y casos de uso

Configuración de los Escenarios

Nombre	Clase	Escenario
Scenery	GraphListTest	Vertex: “A” Vertex: “B” Vertex: “C” Vertex: “D” Vertex: “E”
Scenery	GraphMatrizTest	Vertex: “A” Vertex: “B” Vertex: “C” Vertex: “D” Vertex: “E”

* El nombre de los escenarios puede ser setupStage1, setupStage2, etc.
* La clase es la clase de testing correspondiente al modelo donde acontece el escenario. Por ejemplo si usted está probando User, clase será UserTest.
* El escenario es la descripción de las condiciones iniciales del escenario.

Diseño de Casos de Prueba

Objetivo de la Prueba: check that the bfs methods work

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
DFSTest	TestRouteDFS		Arraylist with vertex and Edges	that all vertices and edges are added correctly
DFSTest	testDFSRouteWithoutEdges		Arraylist with only vertex	that all vertices are added correctly
DFSTest	testDFSGetTrees		Arraylist with vertex and Edges	the count of the number of trees generated is equal to that expected “1”
DFSTest	testDFSGetMoreThanTree		Arraylist with vertex and Edges	the count of the number of trees generated is equal to that expected “2”
DFSTest	graphDFSTreesWithoutEdges		Arraylist with only vertex	the count of the number of trees generated is equal to that expected “4”
DFSTest	testDFSRouteWithNoEdges		Arraylist with only vertex	the added values are the same as expected
DFSTest	testDFSRouteInConnectedGrapgh		Arraylist with vertex and Edges	the added values are the same as expected
DFSTest	testDFSRouteInDisconnectedGrapgh		Arraylist with vertex and Edges	the added values are the same as expected

Objetivo de la Prueba: check that the FloydWarshall methods work

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
FloydWarshallTest	testFloydWarshallList		Arraylist with vertex and Edges	that the expected distance is the one that the program gives us
FloydWarshallTest	testListFloydWarshallDisconnectedGraph		Arraylist with vertex and Edges	that the expected distance is the one that the program gives us
FloydWarshallTest	testFloyWarshall_SingleVertexGraph		Arralist with only one vertex	the distance is 0 since there is no other vertex to go to
FloydWarshallTest	testMatrixShortesPath		Arraylist with vertex and Edges String	that the expected distance is the one that the program gives us
FloydWarshallTest	testMatrixFloydWarshallDisconnectedGragh		Arraylist with vertex	that there is no distance, that is, it returns us infinite in some cases, since there are no edges
FloydWarshallTest	testMatrixFloydWarshallAlgorithmSingleVertex		Arralist with only one vertex	the distance is 0 since there is no other vertex to go to

Objetivo de la Prueba: check that the Kruskal methods work

Clase	Método	Escenario	Valores de Entrada	Resultado esperado
KuskalAlgorithmTest	testListMinimunSpanningTree		Arraylist with vertex and Edges with cost	that the value returned by the program is the same as expected
KuskalAlgorithmTest	testEmptyGraph		ArrayList null	Since there are no vertices, the minimum value is expected to be returned, that is, 0
KuskalAlgorithmTest	testSingVertex		Arraylist with an only vertex	the cost is 0 since there is no other vertex to go to
KuskalAlgorithmTest	testMatrixFindMinimumSpanningTree		Arraylist with vertex and Edges with cost	that the value returned by the program is the same as expected
KuskalAlgorithmTest	Matrix_SingleVertex		Arraylist with an only vertex	the cost is 0 since there is no other vertex to go to
KuskalAlgorithmTest	testFindMinimumSpanningTree_EmptyGraph		ArrayList null	Since there are no vertices, the minimum value is expected to be returned, that is, 0

UNIVERSIDAD ICESI

Algoritmos y Programación II
Facultad de ingeniería

Objetivo de la Prueba: check that the Prim methods work				
Clase	**Método**	**Escenario**	**Valores de Entrada**	**Resultado esperado**
PrimAlgorithmTest	testListMinimumSpanningTree		Arraylist with vertex and Edges with cost	that the target arraylist contains the same edges as the expected arraylist
PrimAlgorithmTest	testListEmptyGraph		ArrayList null	since there are no vertices, then there are no vertices in which to pass
PrimAlgorithmTest	testListSingleVertex		Arraylist with an only vertex	the cost is 0 since there is no other vertex to go to
PrimAlgorithmTest	testMatrixSingleVertexGraph		Arraylist with an only vertex	the cost is 0 since there is no other vertex to go to
PrimAlgorithmTest	testMatrixEmptyGraph		ArrayList null	since there are no vertices, then there are no vertices in which to pass
PrimAlgorithmTest	testMatrixMinimumSpanningTree		Arraylist with vertex and Edges with cost	that the expected elements are equal to those returned by the algorithm

Objetivo de la Prueba: check that the GraphList methods work				
Clase	**Método**	**Escenario**	**Valores de Entrada**	**Resultado esperado**
GraphListTest	testAddVertex	Scenery	Vertex (“H”)	that the input values are equal to what is expected
GraphListTest	testAddVertex 2	Scenery	Vertex(“B”)	that the input values are different from those expected in the test
GraphListTest	testAddVertex 3	Scenery	Vertex(“A”) Vertex(“B”)	that the input values are different from those expected in the test
GraphListTest	testAddEdge	Scenery	Edge(Vertex”A”, Vertex”B”)	that the edge if it has been created
GraphListTest	testAddEdge2	Scenery	Edge(Vertex”A”)	throw an exception in which do not let add the edge
GraphListTest	testAddEdge3	Scenery	Edge(Vertex”H”,vertex”I” , 10,20)	that an exception is not thrown in which the edge is not allowed to be added
GraphListTest	testDijkstra		Arraylist with vertex, and edges.	that the values delivered to the algorithm are the same as those expected from it
GraphListTest	testDijkstra2		Arraylist with vertex, and edges.	return what is expected
GraphListTest	testDijkstra3		Arraylist with vertex, and edges.	return what is expected

GraphListTest	testBfs		Arraylist with vertex, and edges.	have the method return what is expected based on your input
GraphListTest	testBfs2		Arraylist with vertex, and edges.	have the method return what is expected based on your input
GraphListTest	testBfs3		Arraylist with vertex, and edges.	have the method return what is expected based on your input

Objetivo de la Prueba: check that the GraphMatriz methods work

Clase	**Método**	**Escenario**	**Valores de Entrada**	**Resultado esperado**
GraphMatrizTest	testAddVertex	Scenery	Vertex (“H”)	that the input values are equal to what is expected
GraphMatrizTest	testAddVertex 2	Scenery	Vertex(“B”)	that the input values are different from those expected in the test
GraphMatrizTest	testAddVertex 3	Scenery	Vertex(“A”) Vertex(“B”)	that the input values are different from those expected in the test
GraphMatrizTest	testAddEdge	Scenery	Edge(Vertex”A”, Vertex”B”)	that the edge if it has been created
GraphMatrizTest	testAddEdge2	Scenery	Edge(Vertex”A”)	throw an exception in which do not let add the edge
GraphMatrizTest	testAddEdge3	Scenery	Edge(Vertex”H”,vertex”I” , 10,20)	that an exception is not thrown in which the edge is not allowed to be added
GraphMatrizTest	testDijkstra		Arraylist with vertex, and edges.	that the values delivered to the algorithm are the same as those expected from it
GraphMatrizTest	testDijkstra2		Arraylist with vertex, and edges.	return what is expected
GraphListTest	testDijkstra3		Arraylist with vertex, and edges.	return what is expected
GraphMatrizTest	testBfs		Arraylist with vertex, and edges.	have the method return what is expected based on your input
GraphMatrizTest	testBfs2		Arraylist with vertex, and edges.	have the method return what is expected based on your input
GraphMatrizTest	testBfs3		Arraylist with vertex, and edges.	have the method return what is expected based on your input

* Una prueba se compone de un conjunto de casos de prueba.
* Cada fila representa un ***caso de prueba*** difente

* En el objetivo de la prueba debe escribir una descripción sobre qué es lo que específicamente está probando del modelo del programa.
* La clase es la clase del modelo que está siendo puesto a prueba.
* El método es específicamente el método de la clase que está siendo puesto a prueba.
* El escenario se refiere al nombre del escenario que usted definió. Todos los casos de prueba corresponden a escenarios.
* Los valores de entrada son valores que entran al método puesto a prueba.
* El resultado esperado es lo que se espera que suceda luego de ejecutar el método.