unit testing diagram

class	method	stage	Input	output
MAGraph<T>	DFS()	a graph that does not have the time of the vertices, then the method is executed and it is checked if I arrange the times in the correct way	vertices with their connections	added the times well
MAGraph<T>	DFS()	you enter a graph that has some non-connected vertices, the method is executed and the times of each vertex are evaluated, if they are correct it means that the program is working well.	graph	added the times well
MAGraph<T>	BFS(t)	To a graph with 6 vertices, the bfs is asked to calculate the distance from the origin vertex to the other vertices	vertices with their connections	correctly calculated the distances
MAGraph<T>	BFS(t)	you enter a graph that has some	graph	correctly calculated the distances

		non-connected vertices, the method is executed and the distance of each vertex is evaluated, if they are correct it means that the program is working well.		
MAGraph<T>	floydWarshallTest()	a graph is entered and then the method is executed. To verify that the method did it correctly, the solution matrix is compared with the matrix that returned the method	graph	the matrices are the same, so the method worked correctly
MAGraph<T>	floydWarshallTest()	a graph is entered that has some vertices that are not connected to the rest and then the method is executed. To verify that the method did it correctly, the solution matrix is compared to the matrix returned by	graph	the matrices are the same, so the method worked correctly

		the method.		
MAGraph<T>	dijkstraTest()	You enter a graph and the origin node to find the shortest path with the rest of vertices. to know if the method worked, you are asked the minimum distance of a node that we know	graph	calculate the minimum distance well
MAGraph<T>	dijkstraTest()	You enter a graph, which is going to have some unconnected vertices, and the origin node to find the shortest path with the rest of the vertices. to find out if the method worked, it asks the minimum distance of a node that we know	graph	calculate the minimum distance well
MAGraph<T>	kruskalTest()	Enter a graph, you will be asked to calculate the tree with the lightest edges connecting the vertices. To know if it was done	graph	I calculate the minimum distance tree correctly

		correctly, the edges are added and it is evaluated if they are equal to the sum of the minor edges that we know		
MAGraph<T>	kruskalTest()	Enter a graph, which is going to have some unrelated vertices, and you will be asked to calculate the tree with the lighter edges connecting the vertices. To know if it was done correctly, the edges are added and it is evaluated if they are equal to the sum of the minor edges that we know	graph	I calculate the minimum distance tree correctly
ALGraph<T>	DFS()	a graph that does not have the time of the vertices, then the method is executed and it is checked if I arrange the times in the correct way	vertices with their connections	added the times well

ALGraph<T>	DFS()	you enter a graph that has some non-connect ed vertices, the method is executed and the times of each vertex are evaluated, if they are correct it means that the program is working well.	graph	added the times well
ALGraph<T>	BFS(t)	To a graph with 6 vertices, the bfs is asked to calculate the distance from the origin vertex to the other vertices	vertices with their connections	correctly calculated the distances
ALGraph<T>	BFS(t)	you enter a graph that has some non-connect ed vertices, the method is executed and the distance of each vertex is evaluated, if they are correct it means that the program is working well	graph	correctly calculated the distances

ALGraph<T>	floydWarshallTest()	a graph is entered and then the method is executed. To verify that the method did it correctly, the solution matrix is compared with the matrix that returned the method	graph	the matrices are the same, so the method worked correctly
ALGraph<T>	floydWarshallTest()	a graph is entered that has some vertices that are not connected to the rest and then the method is executed. To verify that the method did it correctly, the solution matrix is compared to the matrix returned by the method.	graph	the matrices are the same, so the method worked correctly
ALGraph<T>	dijkstraTest()	You enter a matrix and the origin node to find the shortest path with each pair of vertices. to know if the method worked, you are asked the minimum	graph	calculate the minimum distance well

		distance of a node that we know		
ALGraph<T>	dijkstraTest()	You enter a graph, which is going to have some unconnected vertices, and the origin node to find the shortest path with the rest of the vertices. to find out if the method worked, it asks the minimum distance of a node that we know	graph	calculate the minimum distance well
ALGraph<T>	kruskalTest()	You enter a graph, you are asked to calculate the edges with less weight that connect the vertices. To find out if it was done correctly, the edges are added and it is evaluated if they are equal to the sum of the minor edges that we know	graph	I calculate the minimum distance tree correctly
ALGraph<T>	kruskalTest()	Enter a graph, which is going to have some unrelated	graph	I calculate the minimum distance tree correctly

		vertices, and you will be asked to calculate the tree with the lighter edges connecting the vertices. To know if it was done correctly, the edges are added and it is evaluated if they are equal to the sum of the minor edges that we knowvertices , and you will be asked to calculate the tree with the lighter edges connecting the vertices. To know if it was done correctly, the edges are added and it is evaluated if they are equal to the sum of the minor edges that we know		