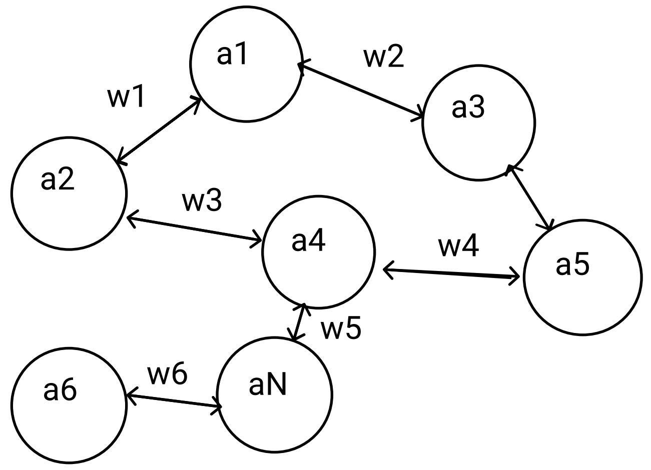
ADT Graph Adjacency Matrix



Invariant
A={ a1, a2, a3, a4,a5,a6...aN}
W={w1,w2,w3,w4,w5,w6...wN}

1. wN>0
2. A != empty
3. No loops

Primitive operations:

-createGraph: ->Graph
-addVertex:Element x numArray1 x numArray1
->Grap
-EnlargeMatrix:Graph ->Graph
-addEdge:num1 x num2 x num3 x Graph ->Graph
-getWeight: num1 x num2 x Graph->Numero
-getConection: Graph ->Vertex
-floydWarshall: Graph ->numArray
-dijkstra: numSource x Graph ->numArray

Graph(adjacencyMatrix, vertexes) : Constructor

“Create a new graph with a determinated matrix and vertexes”

{ pre: true }
{ post: fInstance and create a new graph with a determinated vertexes and matrix }

floydWarshall( ) : Analyzer

“Use the acquaintance floydWarshall method to return a matrix of weights”

{ pre: the graph must be initialized and with its adjacency matrix }
{ post: Return a matrix of calculated weights }

addVertex(element, edges, weigts) : Modifier

“Insert a new vertex inside the graph with a determinated edges and weightss”

{ pre: The graph must be initializated }
{ post: The graph with a new vertex associated }

dijkstra( ) : Analyzer

“Use the acquaintance dijkstra method to return a matrix of mininum weights”

{ pre: The graph must be initialized and with its adjacency matrix }
{ post: Return a matrix of calculated weights }

addEdge(numVertex1, numVertex2, weight) : Modifier

“Insert a new edge inside the graph between two vertex and assigns a weight”

{ pre: The graph must be initializated }
{ post: The graph with a new edge associated with a determinated weight }

getWeight(vertex1, vertvex2) : Analyzer

“Usted to get the weight between two vertex using the adyacency matrix”

{ pre: The graph must be initialized and with its adjacency matrix }
{ post: Return the weight between two vertexes in a int variable }

getConnection(vertex1, vertex2) : Analyzer

“Get the connection between two vertexes and return a Vertex connector”

{ pre: The graph must be initializated and the two vertex must be exists }
{ post: Return the vertex connector between the two vertexes desired }