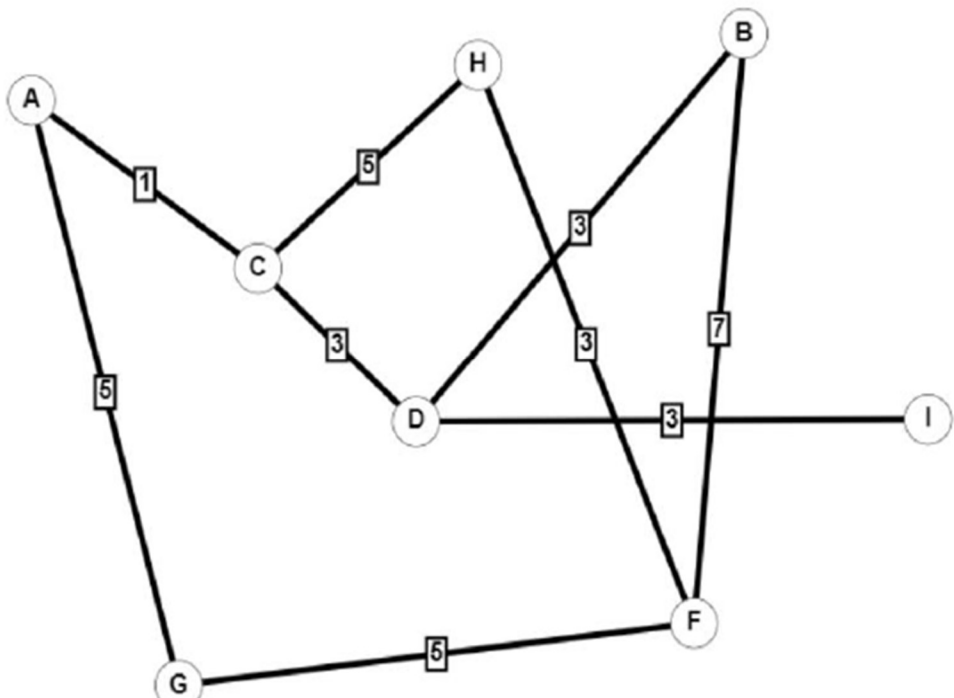


Simple Graph ADT		
$G = (V, E)$  <p> $V = \{A, B, C, D, E, F, H, I\}$ $E = \{(A, G), (A, C), (C, H), (D, B), (D, I), \dots\}$ </p>		
<p>Note: The number on each of the edges is the weight of the respective edge, which can be any value as long as it is an integer.</p> <p>Weight = w</p>		
<p> $inv = \{no\ loops\}$ $inv = \{\forall w \mid w > 0\}$ $inv = \{V \neq \emptyset\}$ $inv = \{E \neq \emptyset\}$ $inv = \{derigged\}$ </p>		
Primitive Operations		
SimpleGraph	...	SimpleGraph
Create Vertex	SimpleGraph x Pos	SimpleGraph x Vertex
Create Edge	SimpleGraph x V1 x V2	SimpleGraph x Edge
Dijkstra Path	SimpleGraph x V1 x V2	SimpleGraph x VertexList
Dijkstra Amount	SimpleGraph x VertexList	Integer
Floyd Warshall	SimpleGraph	SimpleGraph x VertexList

SimpleGraph(): Constructor
Creates a new Simple Graph
Pre = {true}
Pos = {SimpleGraph}

Create Vertex(Pos): Creator
Creates a new Vertex in the Graph, this vertex must have an unique identification
Pre = {true, SimpleGraph}
Pos = {a new vertex without connexion}

Create Edge(V vertex1, V vertex2, W weight) : Creator
Creates a new edge between two vertexes, this edge has a weight and its an integer.
$pre = \{vertex1, vertex2\}$ $pos = \{edge\ between\ vertex1, vertex2\}$

Dijkstra Path(V vertex1, Vertex2) : Analyzer
Visit all the paths between the vertex1(Source) and vertex2, evaluate any paths to know which one is the shortest one between those vertexes, and return a list with the vertexes of the path
$pre = \{true, vertex1, vertex2\}$ $pos = \{list\ with\ vertexes\}$

Floyd Warshall() : Analyzer
Evaluates all paths between each vertex to find the shortest path between each pair of vertexes
$pre = \{true\}$ $pos = \{list\ with\ vertexes\}$