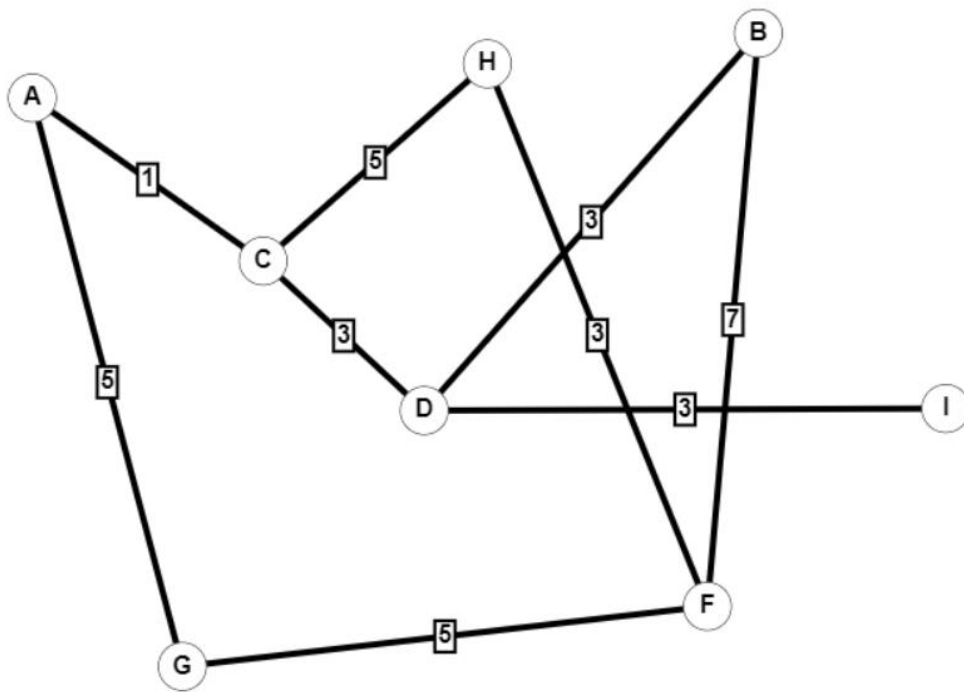


## Simple Graph ADT

$$G = (V, E)$$



$$V = \{A, B, C, D, E, F, G, H, I\}$$

$$E = \{(A, G), (A, C), (C, H), (D, B), (D, I) \dots\}$$

**Note:** The number in the edge are the weight of the edges, these can be any value, but always must be an Integer  
Weight =  $w$

$$inv = \{no\ lops\}$$

$$inv = \{\forall\ w\ \{w\ |\ w > 0\}\}$$

$$inv = \{V \neq \emptyset\}$$

$$inv = \{E \neq \emptyset\}$$

$$inv = \{derigged\}$$

## Primitive Operations

SGraph	...	SGraph
Create Vertex	SGraph x Pos	SGraph x Vertex
Create Edge	SGraph x V1 x V2	SGraph x Edge
Dijkstra Path	SGraph x V1 x V2	SGraph x List Vertex
Dijkstra Amount	SGraph x List Vertex	Integer
Floyd Warshall	SGraph	SGraph x List Vertex

<b>SGraph( ) : Constructor</b>
Creates a new Simple Graph
<i>pre = {true}</i> <i>pos = {SGraph}</i>

<b>Create Vertex(Pos) : Creator</b>
Creates a new vertex in the graph, this vertex must have an identification
<i>pre = {true, MDGraph}</i> <i>pos = {a new vertex without connexion}</i>

<b>Create Edge(V vertex1, V vertex2, W weight) : Creator</b>
Creates a new edge between two vertexes, this edge has a weight, this weight is an integer greater than zero.
<i>pre = {vertex1, vertex2}</i> <i>pos = {edge between vertex1, vertex2}</i>

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

<b>Dijkstra Amount( ) : Analyzer</b>
Get the list of the method Dijkstra Path, and then evaluate the edges between the vertexes of the list and get the total weight of the path.
<i>pre = {Dijkstra Path}</i> <i>pos = {an integer that is the total weight}</i>

<b>Floyd Warshall( ) : Analyzer</b>
Evaluates all paths between all vertices, to find the shortest path between any pair of vertices.
<i>pre = {true}</i> <i>pos = {list with vertexes}</i>