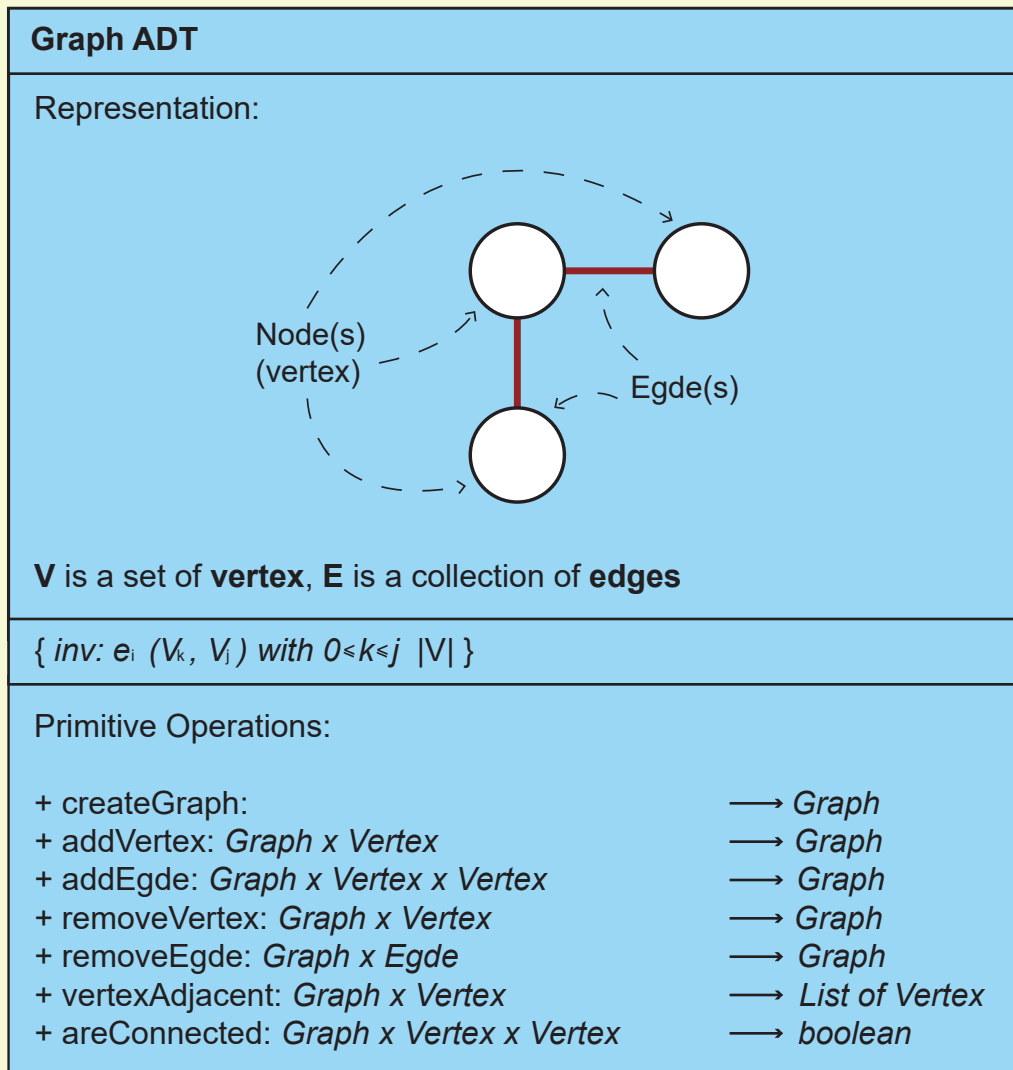


# The ADT Graph

## Definition:

- The **ADT Graph** consists of a finite set of vertices(or nodes) and set of Edges which connect a pair of nodes, together with primitive operations.



### createGraph()

"Creates a new empty graph"

{ pre: TRUE }

{ post: Graph G }

**addVertex(Graph, Vertex)**

"Adds a new vertex to the graph"

{ *pre*: Graph != null ^ Vertex != null }

{ *post*:



}

**addEdge(Graph, Vertex1, Vertex2, e1)**

"Adds a new edge to the graph connecting two existing vertices"

{ *pre*: Vertex1 != null ^ Vertex2 != null ^ Graph != null ^ (Vertex1 ^ Vertex2 ∈ Graph) }



}

{ *post*:

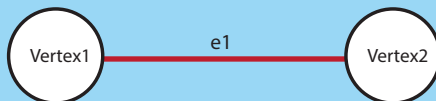


}

**removesVertex(Graph, Vertex2)**

"Removes a given vertex from the graph"

{ *pre*: Vertex1 != null ^ Vertex2 != null ^ Graph != null ^ (Vertex1 ^ Vertex2 ∈ Graph) }



{ *post*:

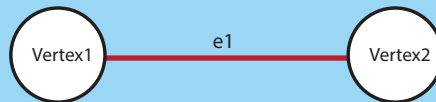


}

### **removesEdge(Graph, e1)**

"Removes a given edge from the graph disconnecting two vertices"

{ pre: Vertex1 != null ^ Vertex2 != null ^ Graph != null ^ (Vertex1 ^ Vertex2 ∈ Graph) ^ e1 != null }



{ post:

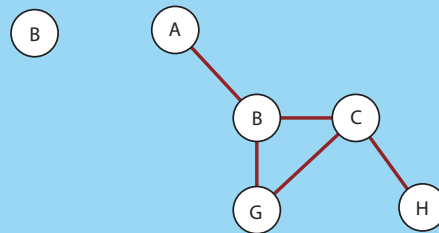


}

### **vertexAdjacent(Graph, B)**

"Returns a list of vertices that contains all the adjacent vertices of a given vertex"

{ pre:



{ post:



### **areConnected(Graph, vertex1, vertex2)**

"Indicates whether two given vertices share an edge or not"

{ pre: Vertex1 != null ^ Vertex2 != null ^ Graph != null ^ (Vertex1 ^ Vertex2 ∈ Graph) }



{ post: if the vertices share an edge it returns TRUE, else it returns FALSE. }