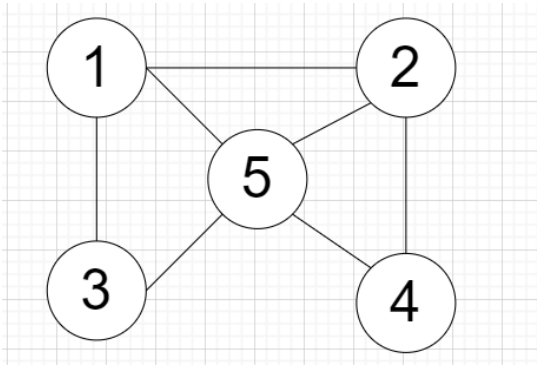
TAD Graph

Graph = {$V = \{v_1, v_2, \ldots, v_n\}$, $E = \{e_1 = (v_{i1}, v_{j1,} w_1), e_2 = (v_{i2}, v_{j2}, w_2), e_m = (v_{im}, v_{jm}, w_m)\}$, directed, weighted}



{inv:

1. $\forall e_k \in E, v_{ik} \in V$ A $vj_k \in V, w_k > 0$
2. $\text{directed} = \text{false} \Rightarrow (\forall (a, b) \in E \; \exists \; (b, a) \in E, a, b \in V)$
3. $\text{weighted} = \text{false} \Rightarrow \forall e_k \in E, w_k = 1$

}

Primitive Operations

- Graph <> → <Graph> Constructor
- insertVertex <Vertex> → <Graph> Modifier
- insertEdge <Vertex, Vertex> → <Graph> Modifier
- delete < Vertex> → <Graph> Modifier
- searchVertex <Graph> → List<Vertex> Analyzer
- dijkstra < Vertex, Vertex> → Analyzer

Operations

Graph (boolean directed, boolean weighted, int n)
Create a new graph that may or may not be directed or weighted.
{pre: }
{post: Graph = {V={}, E={}, directed, weighted }

insertVertex (Vertex v)
Insert a vertex in the graph.
{pre: v ∉ g.V}
{post: v ∈ g.V}

insertEdge (Vertex v1, Vertex v2)
Add an edge of weight 1 that goes from v1 to v2. If the graph is not directed, it also adds it from v2 to v1.
{pre: v1, v2 ∈ g.V }
{post: edge = (v1, v2, 1) ∈ g.E. If g.directed = false, edge = (v2, v1, 1)) ∈ g.E }

insertEdge (Vertex v1, Vertex v2, double weight)
Add an edge of weight 1 that goes from v1 to v2. If the graph is not directed, it also adds it from v2 to v1.
{pre: v1, v2 ∈ g.V, g.weight = true, w > 0}
{post: edge = (v1, v2, weight) ∈ g.E. If g.directed = false, edge = (v2, v1, weight)) ∈ g.E }

delete (Vertex v)
Eliminate v from the graph
{pre: v ∈ g.V }
{post: v ∉ g.V. All vertices that are incidents with v ∉ g. E }

dijkstra (Vertex u, Vertex v)
Carry out the Dijkstra algorithm, taking u as the initial vertex and v as the target vertex
{pre: u ∈ g.V, v ∈ g.V, g}
{post: ∀v ∈ g.V, adds attributes u.pred and u.d, corresponding respectively to the predecessor and the distance added by Dijkstra's algorithm}

search(Vertex v)
Returns if there is a vertex with the given value in the graph.
{pre: }
{post: true if ∃x ∈ g.V : value