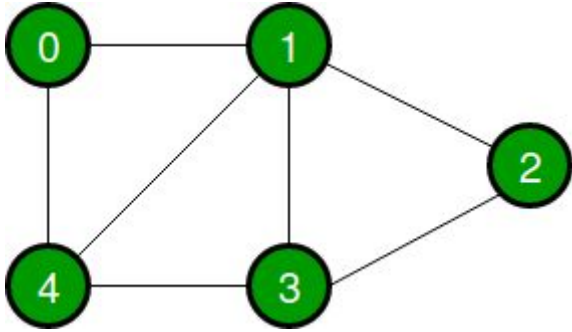


TAD <Graph>																																
																																
{inv: $EdgeQ \Leftarrow VertexQ \wedge Edges\ must\ relate\ two\ vertices\ }$ }																																
Primitive Operations: <table border="0"> <tr> <td><input type="checkbox"/> addVertex:</td><td>Graph x value x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> deleteVertex:</td><td>Graph x value</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> searchVertex:</td><td>Graph</td><td>→ value</td></tr> <tr> <td><input type="checkbox"/> updateVertex:</td><td>Graph x value x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> bfs</td><td>Graph x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> dfs</td><td>Graph x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> dijkstra:</td><td>Graph x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> floydWarshall:</td><td>Graph x vertex</td><td>→ Graph</td></tr> <tr> <td><input type="checkbox"/> prim:</td><td>Graph x vertex</td><td>→ Tree</td></tr> <tr> <td><input type="checkbox"/> kruskal:</td><td>Graph x vertex</td><td>→ Tree</td></tr> </table>			<input type="checkbox"/> addVertex:	Graph x value x vertex	→ Graph	<input type="checkbox"/> deleteVertex:	Graph x value	→ Graph	<input type="checkbox"/> searchVertex:	Graph	→ value	<input type="checkbox"/> updateVertex:	Graph x value x vertex	→ Graph	<input type="checkbox"/> bfs	Graph x vertex	→ Graph	<input type="checkbox"/> dfs	Graph x vertex	→ Graph	<input type="checkbox"/> dijkstra:	Graph x vertex	→ Graph	<input type="checkbox"/> floydWarshall:	Graph x vertex	→ Graph	<input type="checkbox"/> prim:	Graph x vertex	→ Tree	<input type="checkbox"/> kruskal:	Graph x vertex	→ Tree
<input type="checkbox"/> addVertex:	Graph x value x vertex	→ Graph																														
<input type="checkbox"/> deleteVertex:	Graph x value	→ Graph																														
<input type="checkbox"/> searchVertex:	Graph	→ value																														
<input type="checkbox"/> updateVertex:	Graph x value x vertex	→ Graph																														
<input type="checkbox"/> bfs	Graph x vertex	→ Graph																														
<input type="checkbox"/> dfs	Graph x vertex	→ Graph																														
<input type="checkbox"/> dijkstra:	Graph x vertex	→ Graph																														
<input type="checkbox"/> floydWarshall:	Graph x vertex	→ Graph																														
<input type="checkbox"/> prim:	Graph x vertex	→ Tree																														
<input type="checkbox"/> kruskal:	Graph x vertex	→ Tree																														

addVertex
Adds a vertex to the graph as a relation with a given vertex.
pre: Graph is already initialized.
pos: A value was added to the graph adjacent/related to the given vertex.

deleteVertex
Deletes a vertex from the graph, making the necessary relation fixes on its relations
pre: Graph is already initialized.
pos: A value was deleted from the

graph

searchVertex

Search a vertex with a given key in the graph.
--

pre: Graph is already initialized.

pos: vertex value if it was found on the graph, null if it wasn't.
--

updateVertex

Searches the vertex with the given key and, if found, changes its value to the new one.

pre: Graph is already initialized.

pos: new vertex value if it was found on the graph and replaced, null if it wasn't.

BFS

Travels the graph by breadth. Visiting each neighbour vertices before going deeper, until all vertices are visited.

pre: Graph is already initialized.

pos: List showing the vertices of the graph.
--

DFS

Travels the graph by depth. Visiting each relation of each vertex before going wider, until all vertices are visited.

pre: Graph is already initialized.

pos: List showing the vertices of the graph.
--

Dijkstra
Travels the graph from a given vertex returning the minimum path to each of them
pre: Graph is already initialized.
pos: List showing the minimum paths.

Floyd-Warshall
Travels the graph from a given vertex returning the minimum path to each pair.
pre: Graph is already initialized.
pos: Length of shortest paths between all vertices.

Prim
Travels the graph from a given vertex returning the minimum spanning tree including every vertex.
pre: Graph is already initialized.
pos: A tree with every vertex of the graph.

Kruskal
Travels the graph from a given vertex returning the minimum spanning forest including every vertex.
pre: Graph is already initialized.
pos: A forest with every vertex.