

ADT < MATRIX GRAPH >
< vertexIndexMap, vertices, adjacencyMatrix >
Graph (vertexIndexMap \in Map of vertices to indices, vertices \in List of vertices, adjacencyMatrix \in Edge[][])
<ul style="list-style-type: none"> * Primitive Operations: * addVertex $\rightarrow V \rightarrow \text{Void}$ * addEdge $\rightarrow V \times V \rightarrow \text{Void}$ * dfs $\rightarrow V \rightarrow \text{List}<\text{String}>$ * dijkstra $\rightarrow V \rightarrow \text{Map}<V, \text{Integer}>$ * updateMatrix $\rightarrow () \rightarrow \text{Void}$ * getEdge $\rightarrow V \times V \rightarrow \text{Edge}<V>$

Operaciones:

```
addVertex(vertex)
"Adds a vertex to the adjacency matrix"

{Pre: vertex  $\in V$ }

{Post: Adds the vertex to the adjacency matrix}
```

```
addEdge(from, to)
"Adds an edge from 'from' to 'to' in the adjacency matrix"

{Pre: from, to  $\in V$ }

{Post: Adds an edge from 'from' to 'to' in the adjacency matrix}
```

```
dfs(start)
"Performs Depth-First Search (DFS) starting from the specified vertex"

{Pre: start  $\in V$ }

{Post: Returns a list of vertices visited in DFS order}
```

```
dijkstra(start)
"Performs Dijkstra's algorithm starting from the specified vertex"

{Pre: start  $\in V$ }

{Post: Returns a map of vertices and their respective distances from the starting vertex}
```

updateMatrix()

"Updates the adjacency matrix when a new vertex is added"

{Pre: None}

{Post: Updates the adjacency matrix to include the new vertex}

getEdge(from, to)

"Gets the edge from 'from' to 'to' in the adjacency matrix"

{Pre: from, to $\in V$ }

{Post: Returns the edge from 'from' to 'to' or null if no edge exists}