

Graph Class Documentation

Class: Graph

Methods:

`__init__(self, size)`

- Initializes a graph with a given number of vertices.
- **Parameters:**
 - `size` (int): The number of vertices in the graph.

`get_number_of_vertices(self)`

- Returns the number of vertices in the graph.
- **Returns:**
 - int: The number of vertices.

`parse_vertices(self)`

- Returns an iterator over the vertices in the graph.
- **Returns:**
 - iterator: An iterator over the vertices.

`add_vertex(self)`

- Adds a new vertex to the graph.

`remove_vertex(self, vertex)`

- Removes a vertex from the graph.
- **Parameters:**

- `vertex` (int): The index of the vertex to be removed.

`read_from_file(self, file_name)`

- Reads a graph from a file.

- **Parameters:**

- `file_name` (str): The filename of the file to read from.

`write_to_file(self, file_name)`

- Writes the graph to a file.

- **Parameters:**

- `file_name` (str): The name of the file to write to.

`add_edge(self, u, v, w)`

- Adds an edge to the graph.

- **Parameters:**

- `u` (int): The first vertex.
- `v` (int): The second vertex.
- `w` (int): The weight of the edge.

`remove_edge(self, edge_id)`

- Removes an edge from the graph.

- **Parameters:**

- `edge_id` (int): The edge id to be removed.

`get_edge_id(self, u, v)`

- Returns the edge id of an edge.

- **Parameters:**

- `u` (int): The first vertex.
- `v` (int): The second vertex.
- ****Returns:****
 - int: The edge id.

`get_edge_endpoints(self, edge_id)`

- Returns the endpoints of an edge by edge id.
- ****Parameters:****
 - `edge_id` (int): The edge id.
- ****Returns:****
 - tuple: The endpoints of the edge.

`get_edge_by_edge_id(self, edge_id)`

- Returns the edge by edge id.
- ****Parameters:****
 - `edge_id` (int): The edge id.
- ****Returns:****
 - tuple: The edge details.

`set_edge_info(self, edge_id, new_cost)`

- Sets the cost of an edge.
- ****Parameters:****
 - `edge_id` (int): The edge id.
 - `new_cost` (int): The new cost of the edge.

`_is_vertex(self, vertex)`

- Checks if a vertex is valid.

- **Parameters:**
 - ``vertex` (int)`: The vertex index.
- **Returns:**
 - bool: True if the vertex is valid, False otherwise.

``get_in_degree(self, vertex)``

- Returns the in-degree of a vertex.
- **Parameters:**
 - ``vertex` (int)`: The vertex index.
- **Returns:**
 - int: The in-degree of the vertex.

``get_out_degree(self, vertex)``

- Returns the out-degree of a vertex.
- **Parameters:**
 - ``vertex` (int)`: The vertex index.
- **Returns:**
 - int: The out-degree of the vertex.

``parse_outbound_edges(self, vertex)``

- Returns an iterator over the outbound edges of a vertex.
- **Parameters:**
 - ``vertex` (int)`: The vertex index.
- **Returns:**
 - iterator: An iterator over the outbound edges.

``parse_inbound_edges(self, vertex)``

- Returns an iterator over the inbound edges of a vertex.
- **Parameters:**
 - `vertex` (int): The vertex index.
- **Returns:**
 - iterator: An iterator over the inbound edges.

`copy_graph(self)`

- Returns a copy of the graph.
- **Returns:**
 - Graph: A copy of the graph.

`bfs(self, start, end)`

- Performs a breadth-first search from start to end.
- **Parameters:**
 - `start` (int): The starting vertex.
 - `end` (int): The ending vertex.
- **Returns:**
 - list: A list with the path from start to end, empty if no path found.

`generate_random_graph(self, number_of_vertices, number_of_edges)`

- Generates a random graph.
- **Parameters:**
 - `number_of_vertices` (int): The number of vertices.
 - `number_of_edges` (int): The number of edges.

`get_number_of_strongly_connected_components(self)`

- Returns the number of strongly connected components in the graph.

- ****Returns:****

- int: The number of strongly connected components.

`_dfs_sct(self, vertex, visited, onStack, ids, low, id)`

- Helper method for depth-first search for strongly connected components.

- ****Parameters:****

- `vertex` (int): The vertex index.

- `visited` (list): List of visited vertices.

- `onStack` (list): List of vertices on the stack.

- `ids` (list): List of vertex ids.

- `low` (list): List of low-link values.

- `id` (list): List containing the current id.

`__str__(self)`

- Returns a string representation of the graph.

- ****Returns:****

- str: A string representation of the graph.

You can now copy this text and paste it into a PDF document.