Strongly connected components documentation

make_directed_graph_from_vertex_list

This method takes a graph object of type DirectedGraph and a vertex_list as input parameters, and returns a new graph object of type DirectedGraph that only contains the vertices in vertex_list, and the edges between them. The new graph is a directed graph, and has the same edge weights as the original graph where applicable.

dfs_out

This method performs a depth-first search (DFS) starting from the given vertex in the graph object of type DirectedGraph. It recursively visits all the vertices that can be reached from vertex by following outgoing edges in the graph. The visited vertices are added to the visited_vertices set, and the vertices are added to a stack dfs_stack in the order they are visited.

dfs_in

This method performs a DFS starting from the given vertex in the graph object of type DirectedGraph. It recursively visits all the vertices that can reach vertex by following incoming edges in the graph. The visited vertices are added to the visited_vertices set, and the vertices are added to a list connected_component in the order they are visited. The method returns connected_component.

get_strongly_connected_components

This method takes a graph object of type DirectedGraph as input, and returns a list of strongly connected components of the graph. It uses the Kosaraju's algorithm to find the strongly connected components. The algorithm consists of two DFS passes over the graph. In the first pass, the vertices are visited in the order they are added to a stack nodes_stack. In the second pass, the vertices are visited in the reverse order they were added to nodes_stack. The method returns a list of new graph objects of type DirectedGraph, where each graph is a strongly connected component of the input graph.

Connected components documentation

dfs

This method performs a depth-first search (DFS) starting from the given vertex in the graph object. It recursively visits all the vertices that can be reached from vertex by following outgoing edges in the graph. The visited vertices are added to the visited_vertices set, and the vertices are added to a list connected_component in the order they are visited. The method returns connected_component.

get_connected_components

This method takes a graph object as input, and returns a list of connected components of the graph. It uses a DFS algorithm to find the connected components. The algorithm iterates over all vertices in the graph, and if a vertex has not been visited yet, it performs a DFS on it to find the connected component that includes the vertex. The method returns a list of new graph objects of type UndirectedGraph, where each graph is a connected component of the input graph.