|  |  |  |
| --- | --- | --- |
| Name of Graph Class’s Public Function in lab.ml | Utility of These Functions | Arguments of These Functions |
| Graph | Constructor to create an empty graph. | (bool) weighted |
| Graph | Constructor to create an empty graph. | (bool) weighted, (bool) directed |
| Graph | Constructor to create a random, connected graph. | (bool) weighted, (int) numVertices, (unsigned long) seed |
| vector<Vertex> getAdjacent | Gets all adjacent vertices to the parameter vertex. Return a vector of vertices | @param  source – vertex to get neighbors from |
| Vertex getStartingVertex() | Returns one vertex in the graph. This function can be used to find a random vertex with which to start a traversal. Return a vertex from the graph |  |
| vector<Vertex> getVertices() | Gets all vertices in the graph. Return a vector of all vertices in the graph |  |
| Edge getEdge | Gets an edge between two vertices Return: if exist, return the corresponding edge; if edge doesn't exist, return Edge(). | @param source - one vertex the edge is connected to  @param destination - the other vertex the edge is connected to |
| vector<Edge> getEdges | Gets all the edges in the graph, return a vector of all the edges in the graph |  |
| bool vertexExists | Checks if the given vertex exists.  return - if Vertex exists, true  if Vertex doesn't exist, return false | (vertex) v |
| bool edgeExists | Checks if edge exists between two vertices exists.  return - if Edge exists, true  - if Edge doesn't exist, return false | (Vertex) source,  (Vertex) destination |
| Edge setEdgeLabel | Sets the edge label of the edge between vertices u and v.  Return - if edge exists, set the label to the corresponding edge(if not directed, set the reverse one too), return edge with new label  - if edge doesn't exist, return InvalidEdge | (Vertex) source, (Vertex) destination, (string) label |
| string getEdgeLabel | Gets the edge label of the edge between vertices u and v.  Return - if edge exists, return edge label  - if edge doesn't exist, return InvalidLabel | (Vertex) source, (Vertex) destination |
| int getEdgeWeight | Gets the weight of the edge between two vertices.  Return - if edge exists, return edge wright  - if doesn't, return InvalidWeight | (Vertex) source, (Vertex) destination |
| void insertVertex | Inserts a new vertex into the graph and initializes its label as "" | (Vertex) v |
| Vertex removeVertex | Removes a given vertex from the graph.  Return - if v exists, return v  \* - if not, return InvalidVertex; | (Vertex) v |
| bool insertEdge | Inserts an edge between two vertices.  \* A boolean is returned for use with the random graph generation.  \* Hence, an error is not thrown when it fails to insert an edge.  Return whether inserting the edge was successful. | (Vertex) source, (Vertex) destination |
| Edge removeEdge | Removes the edge between two vertices.  Return - if edge exists, remove it and return removed edge  - if not, return InvalidEdge | (Vertex) source, (Vertex) destination |
| Edge setEdgeWeight | Sets the weight of the edge between two vertices.  Return - if edge exists, set edge weight and return edge with new weight  - if not, return InvalidEdge | (Vertex) source, (Vertex) destination  (int) weight |
| void initSnapshot | Creates a name for snapshots of the graph. | (string) title |
| void snapshot() | Saves a snapshot of the graph to file.  initSnapshot() must be run first. |  |
| void print() | Prints the graph to stdout. |  |
| void savePNG | Saves the graph as a PNG image. | (string) title |
| bool isDirected() | Judge the boolean output of directed |  |
| void clear() | Clear the adjacency list |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |