Graph and Search Library Tutorial

Andrew Grant, Anton Igorevich, Somya Vasudevan

1 How to Use the Library

- The most important thing is for the user to define his/her vertex and edge data types. The only requirements are that the vertex/edge are comparable (aka must implement operator==) and hashable (aka implement struct hash ...).
- Then the user should select one of `graph_dg`, `graph_dag`, `graph_dt`, `matrix_graph` and provide the struct with two template parameters that specify the vertex and edge types (as mentioned above the lib provides vertex and edge for this but the user can use his/her own data types) e.g. `dag_graph¡my_vertex_1`, `my_edge_1¿ my_graph`; e.g. `dt_graph¡`vertex, edge¿ `my_graph`; e.g. `dg_graph¡my_vertex_2`, `my_edge_2¿ my_graph`;
- At this point any/all of the functions can be used. Note that all functions require pointers as inputs (more specifically `shared_ptrs`); this is to avoid the cost of copying large graphs/vertices/edges see examples/ directory for some examples
- Note that the same function name is used for all graph types, vertex types and edge types. This is another benefit of concepts; that is, concepts are used to make sure the right function is called using overloading