

Report OOP Lab - 20212

Team member group 3:

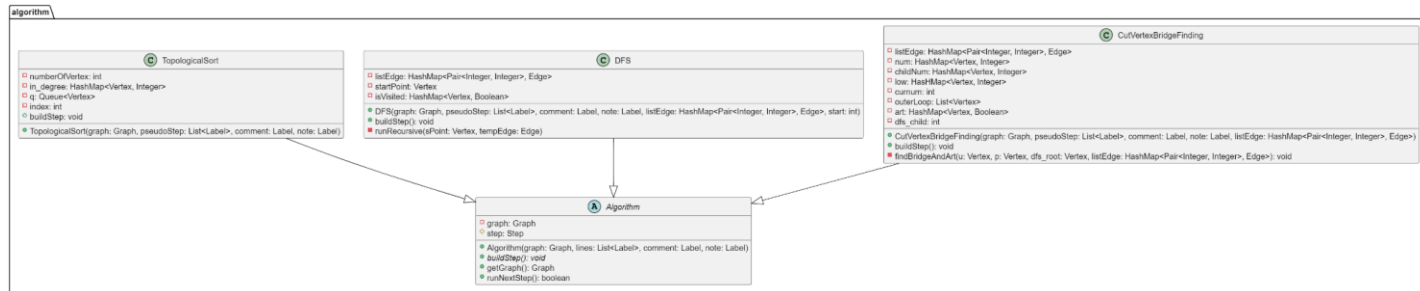
- Nguyễn Thụ Hiếu
- Nguyễn Minh Tuấn
- Trần Tấn Dũng

More information about the source code and high resolution picture can be viewed in the github link of our team:

<https://github.com/hieucuopbien123/ProjectMidtermOOPLab.git>

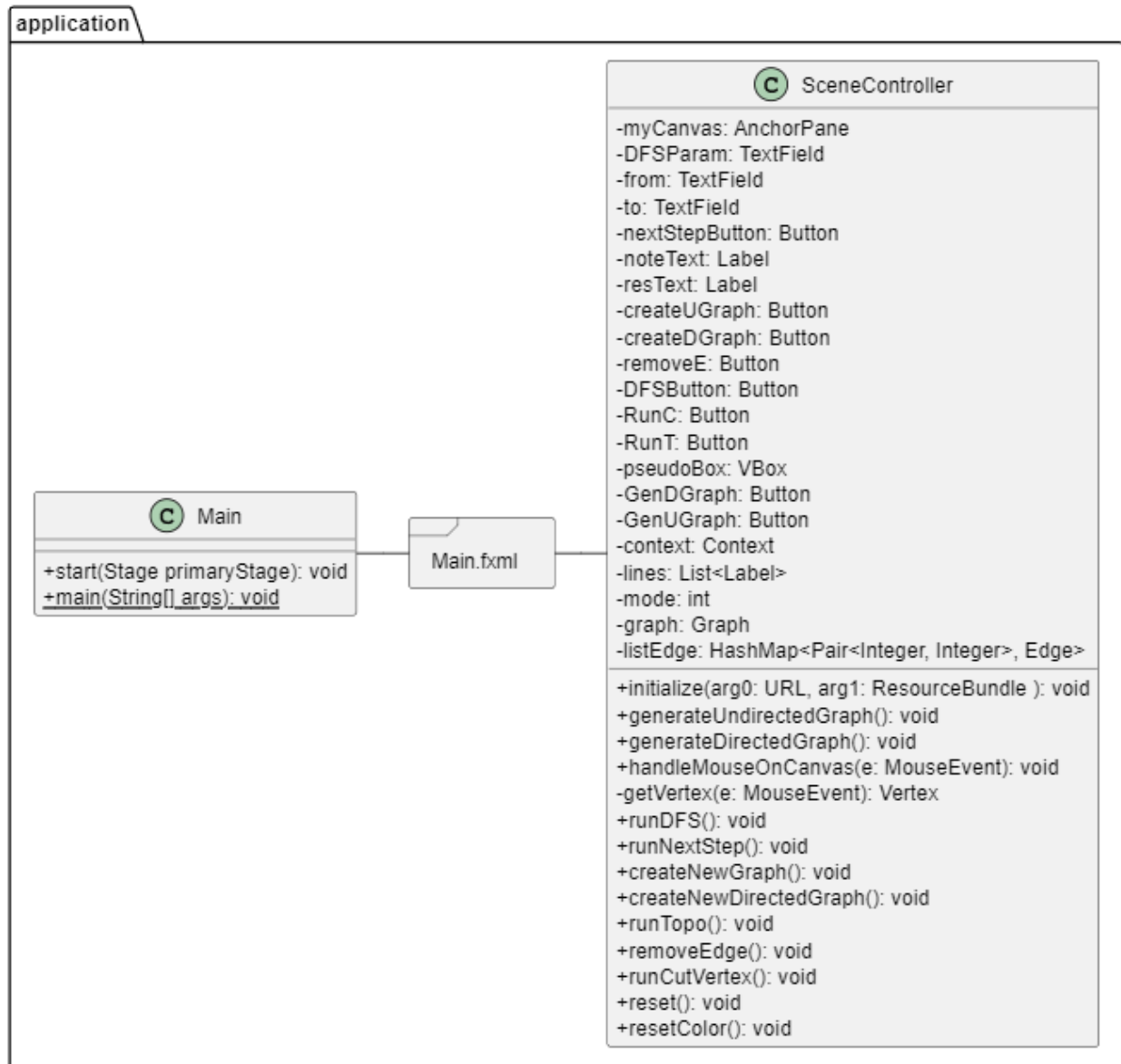
I) Design

- Algorithm package:

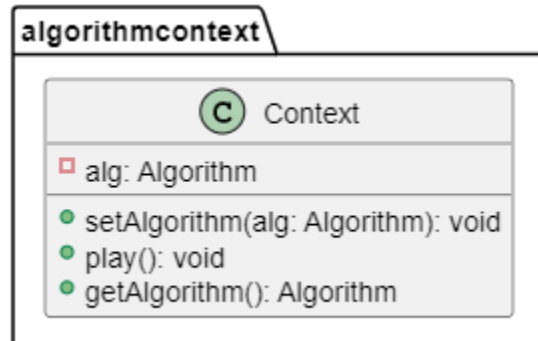


=> Algorithm is an abstract class that is inherited by 3 classes DFS, TopologicalSort and CutVertexandBridgeFinding. The three classes implement the override function buildStep().

- Application package:

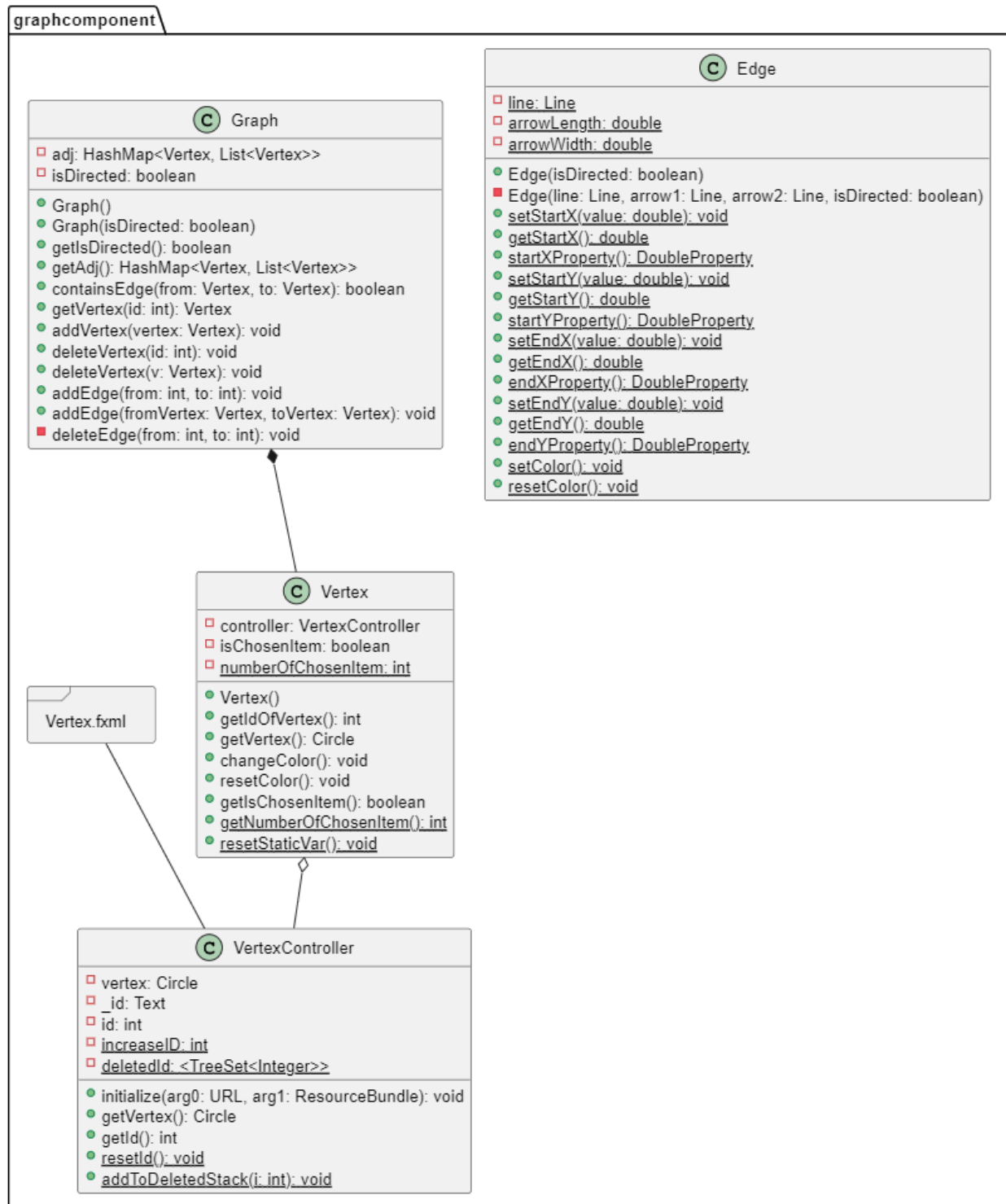


- Context package:



=> Context class aggregates to the Algorithm class will play the role of running the algorithm to build each step and then save to the Step stacks.

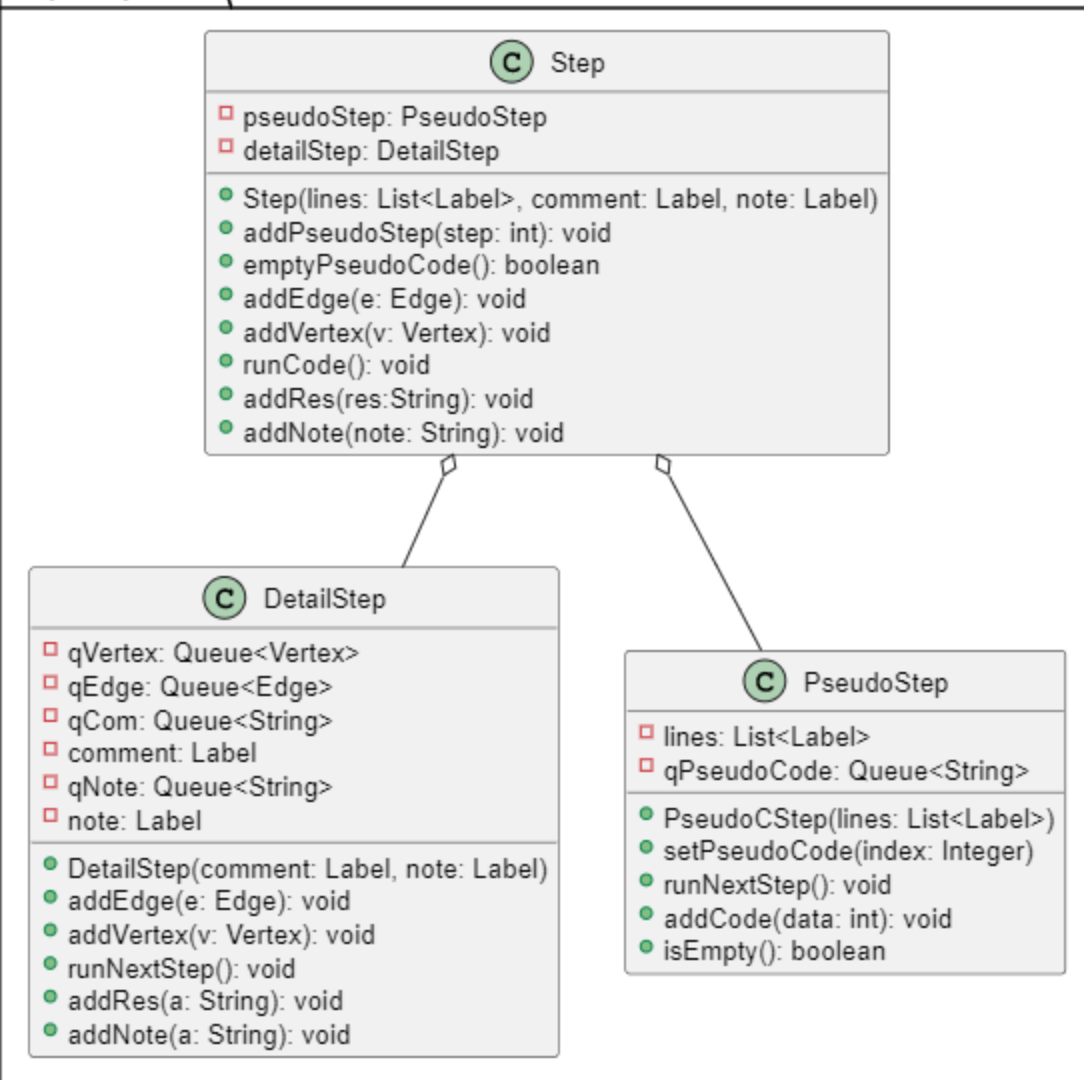
- Graph package:



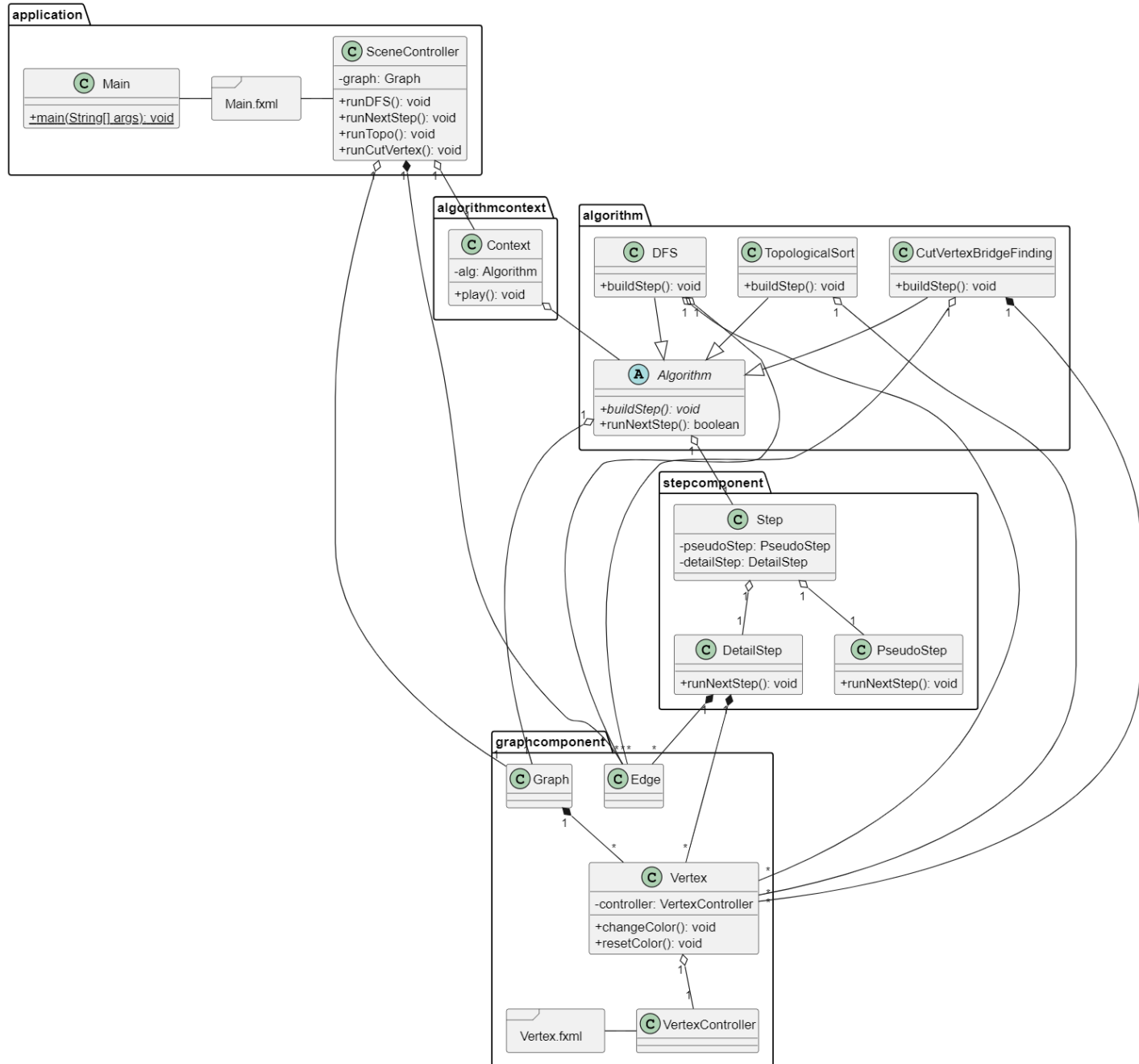
=> Here we only use adjacent list so that Graph doesn't aggregate to Edge class.

- Step package:

stepcomponent



=> Class diagram:



II) Test

1) Build graph:

- Users can build a graph of their own and run algorithms on that particular graph via 2 buttons:

Create a new undirected graph

Create a new directed graph

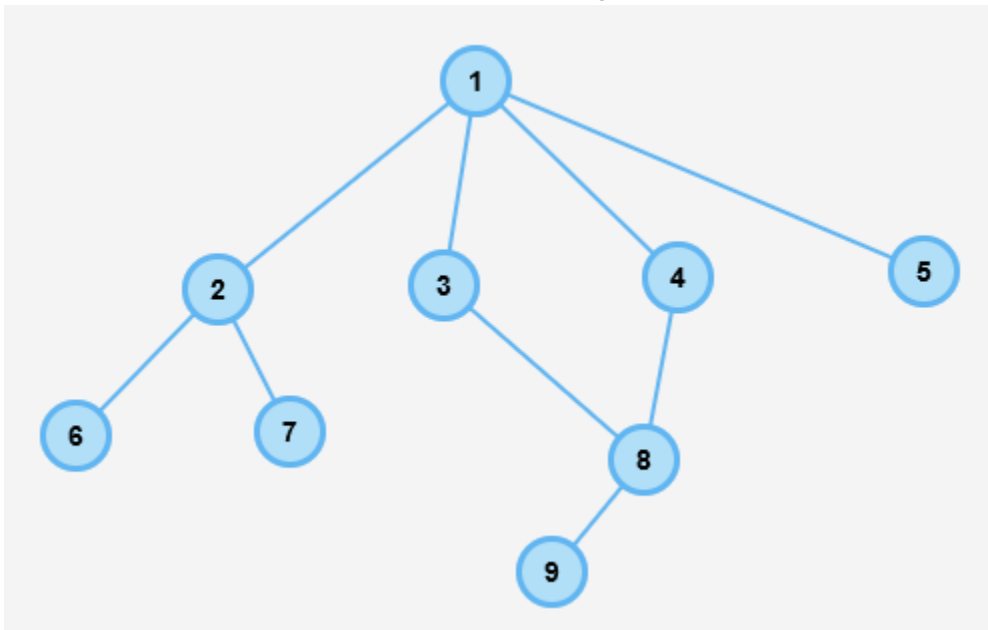
- Users can use example graphs and modify them via 2 button:

Generate Undirected Graph

Generate Directed Graph

Here we can only create 2 example graphs:

- a) Simple connected undirected graph contains cycle:



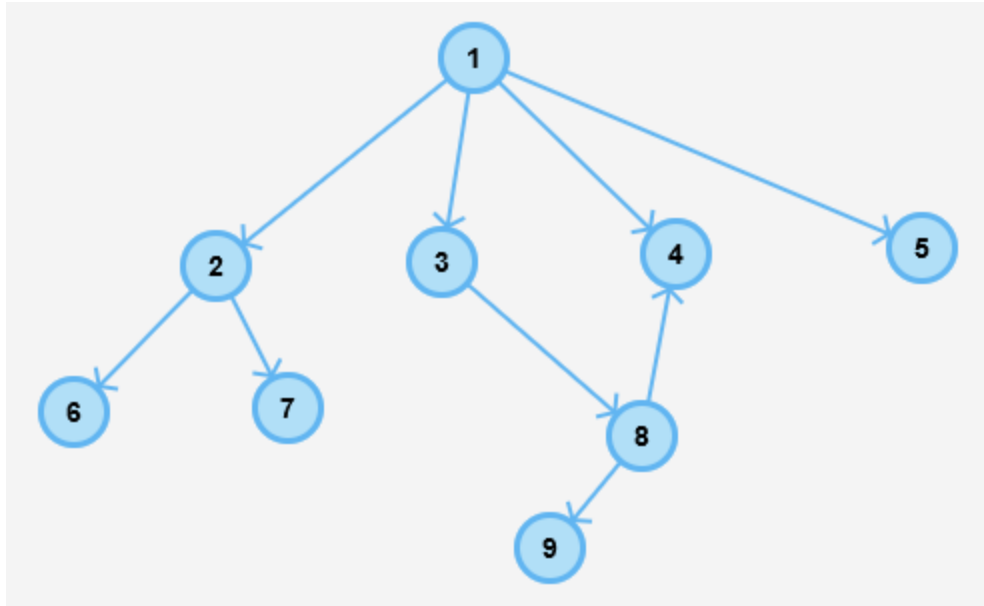
=> This graph can be used for DFS and FindCutVertexAndBridge with results:

DFS: 1 -> 2 -> 6 -> 7 -> 3 -> 8 -> 4 -> 9 -> 5

Cut vertex: 1, 2, 8

Bridge: 2 - 6, 2 - 7, 1 - 2, 8 - 9, 1 - 5

- b) Simple weakly connected directed graph does not contain cycle:



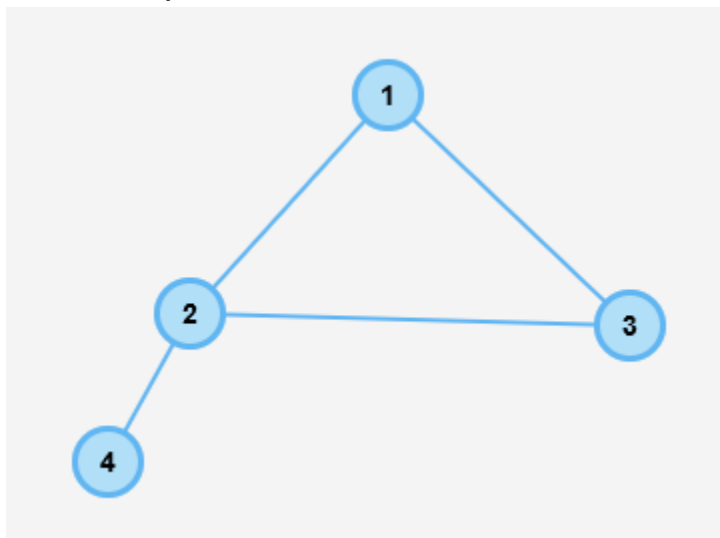
=> This graph doesn't contain cycle, so it can be used for topological sort with the result:

1 -> 2 -> 3 -> 5 -> 6 -> 7 -> 8 -> 4 -> 9

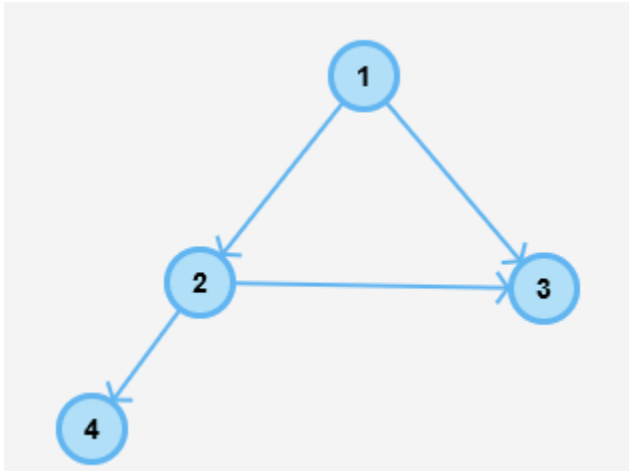
2) Algorithm

Complexity with V is the number of vertices and E is the number of edges, all three algorithms have the time complexity of $O(V+E)$. We would consider three example graphs to test the execution time of each algorithm.

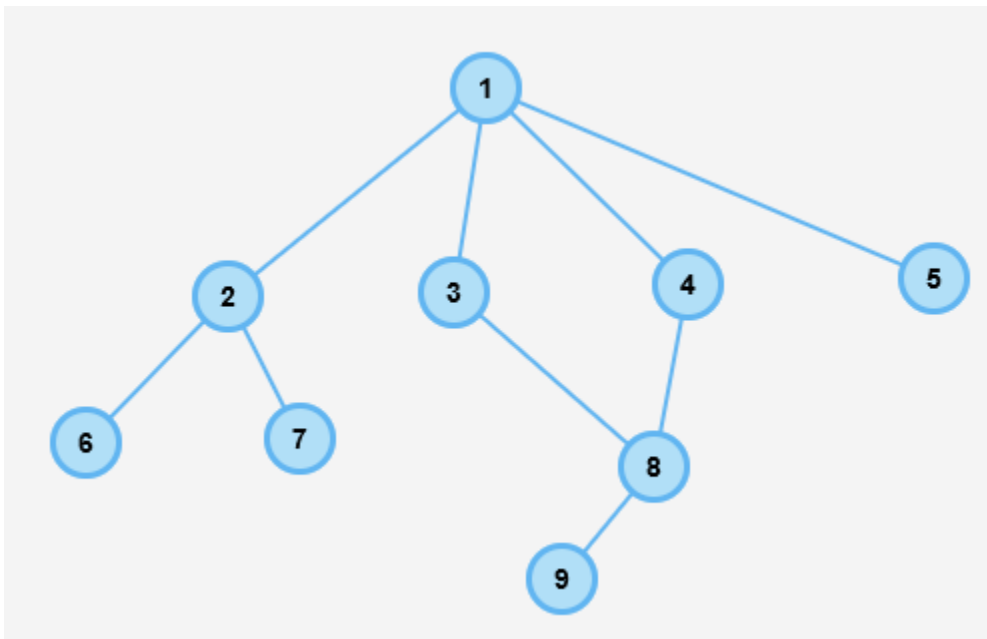
- Graph 1: $V = 4$, $E = 4$



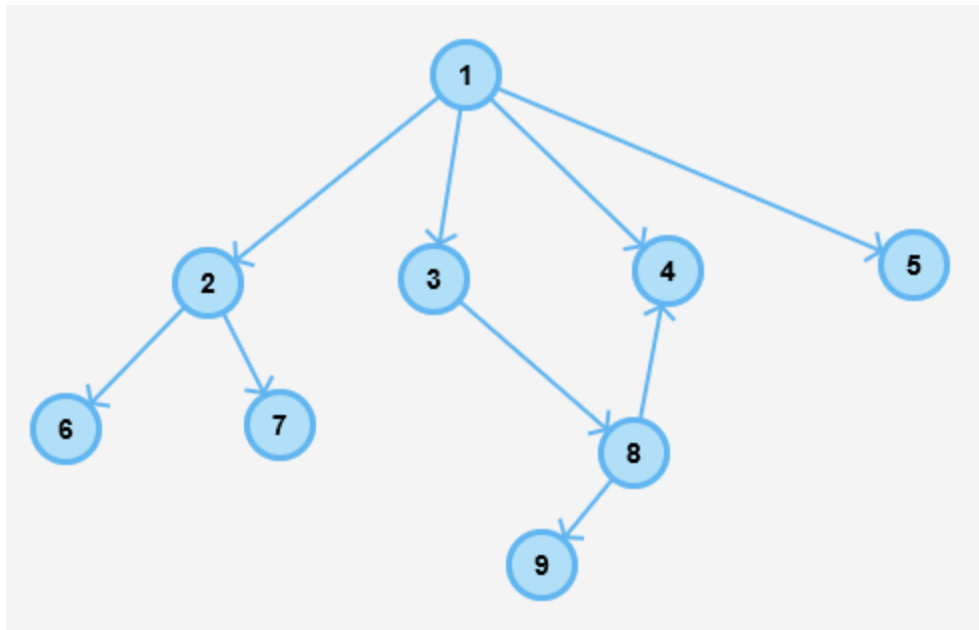
and



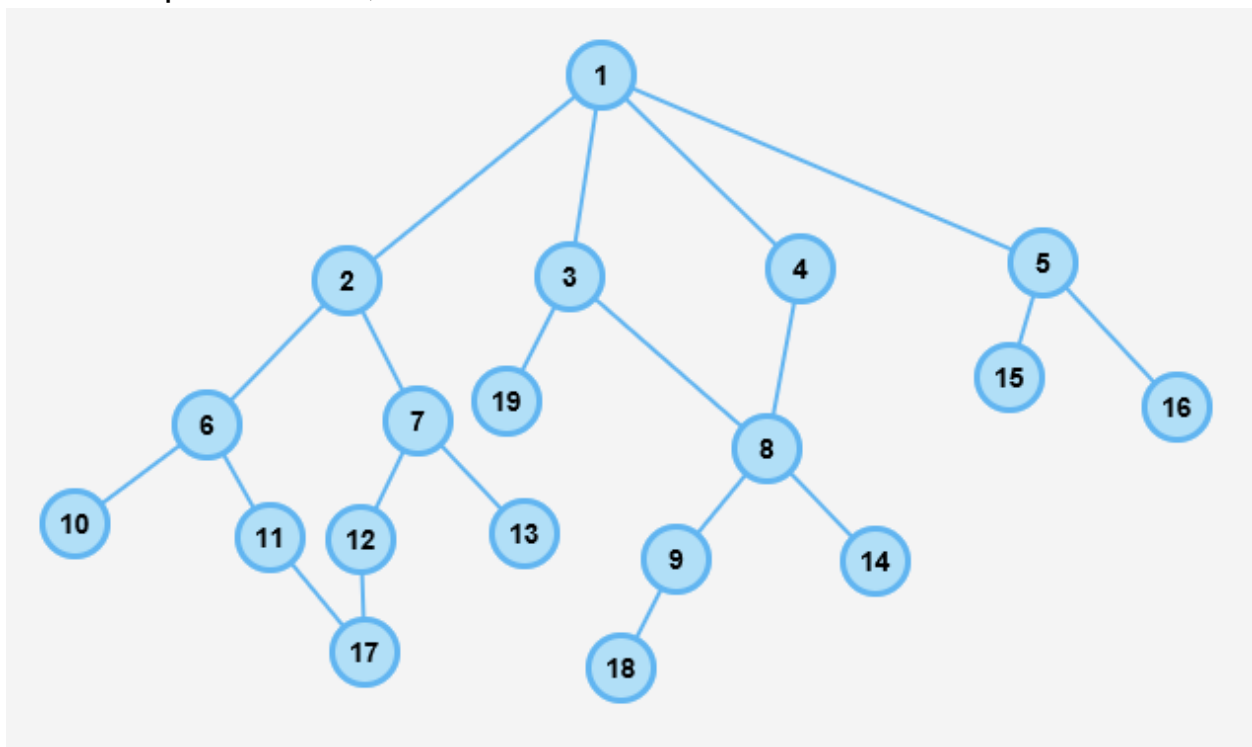
- Graph 2: $V = 9$, $E = 9$



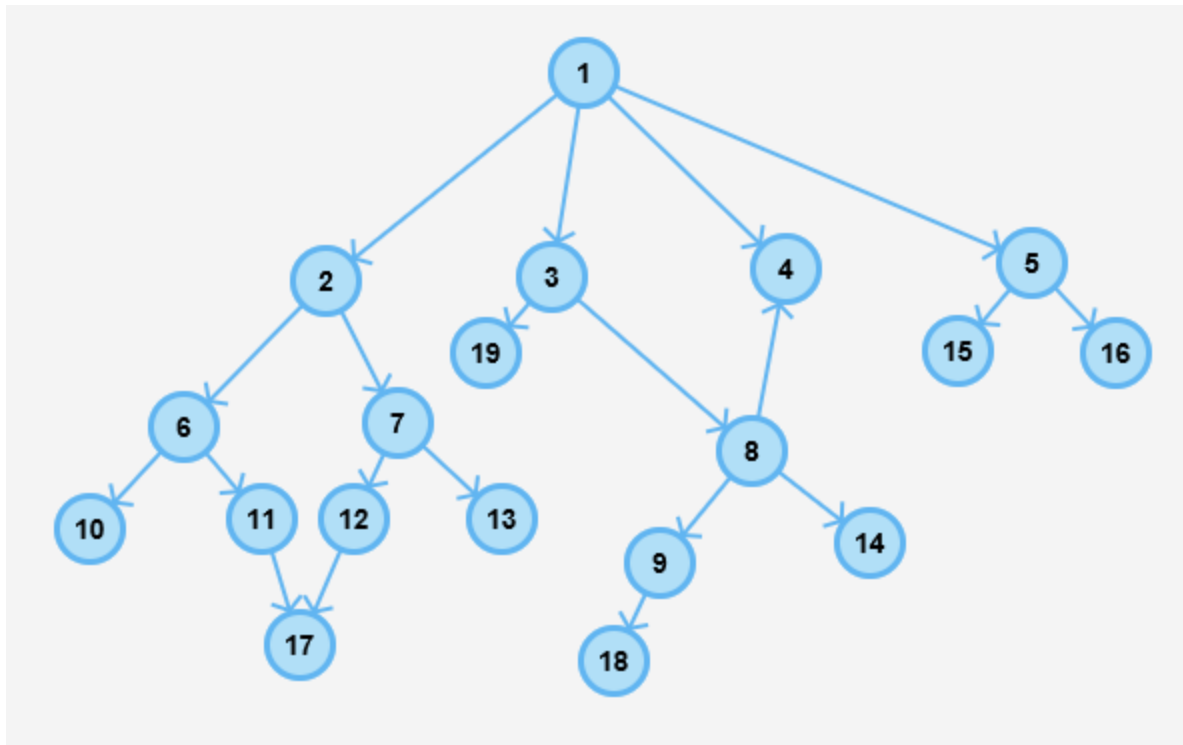
and



- Graph 3: $V = 19$, $E = 20$



and



- Benchmark table:

	DFS	Cut vertex and bridge finding	Topological sort
Step graph 1	25	35	25
Time graph 1	0.171ms	0.175ms	0.143ms
Step graph 2	56	76	56
Time graph 2	0.38ms	0.5ms	0.23ms
Step graph 3	122	185	115
Time graph 3	0.385ms	0.502ms	0.306ms