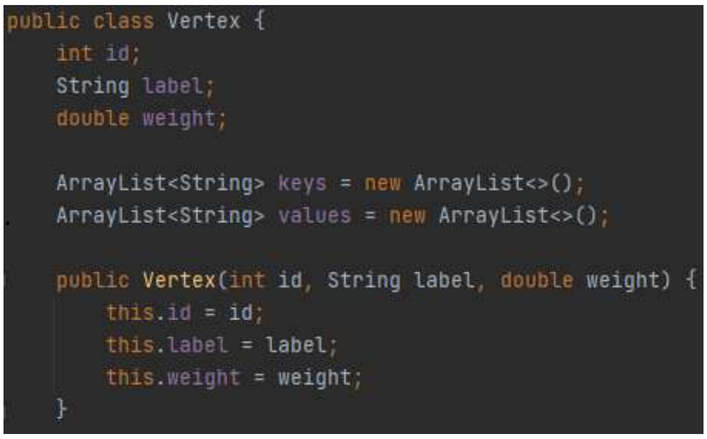
GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework 8 Report

Tuba Toprak
161044116

1. SYSTEM REQUIREMENTS

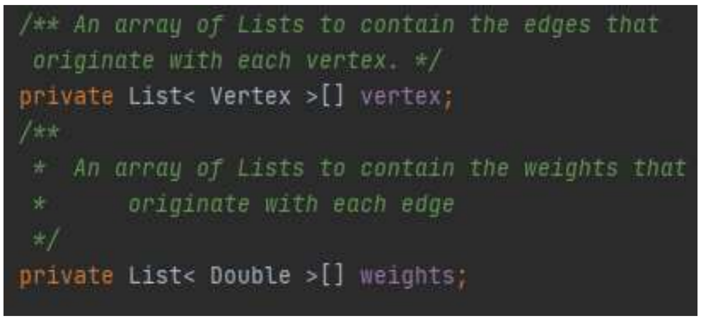
As far as I understood from the homework report, the adjacency List should be made with vertex classes, not edges**.** For this, I created a vertex class in the requested format.



```java
public class Vertex {
    int id;
    String label;
    double weight;

    ArrayList<String> keys = new ArrayList<>();
    ArrayList<String> values = new ArrayList<>();

    public Vertex(int id, String label, double weight) {
        this.id = id;
        this.label = label;
        this.weight = weight;
    }
```

Since a vertex will have multiple key-values, I have stored them in an arraylist.

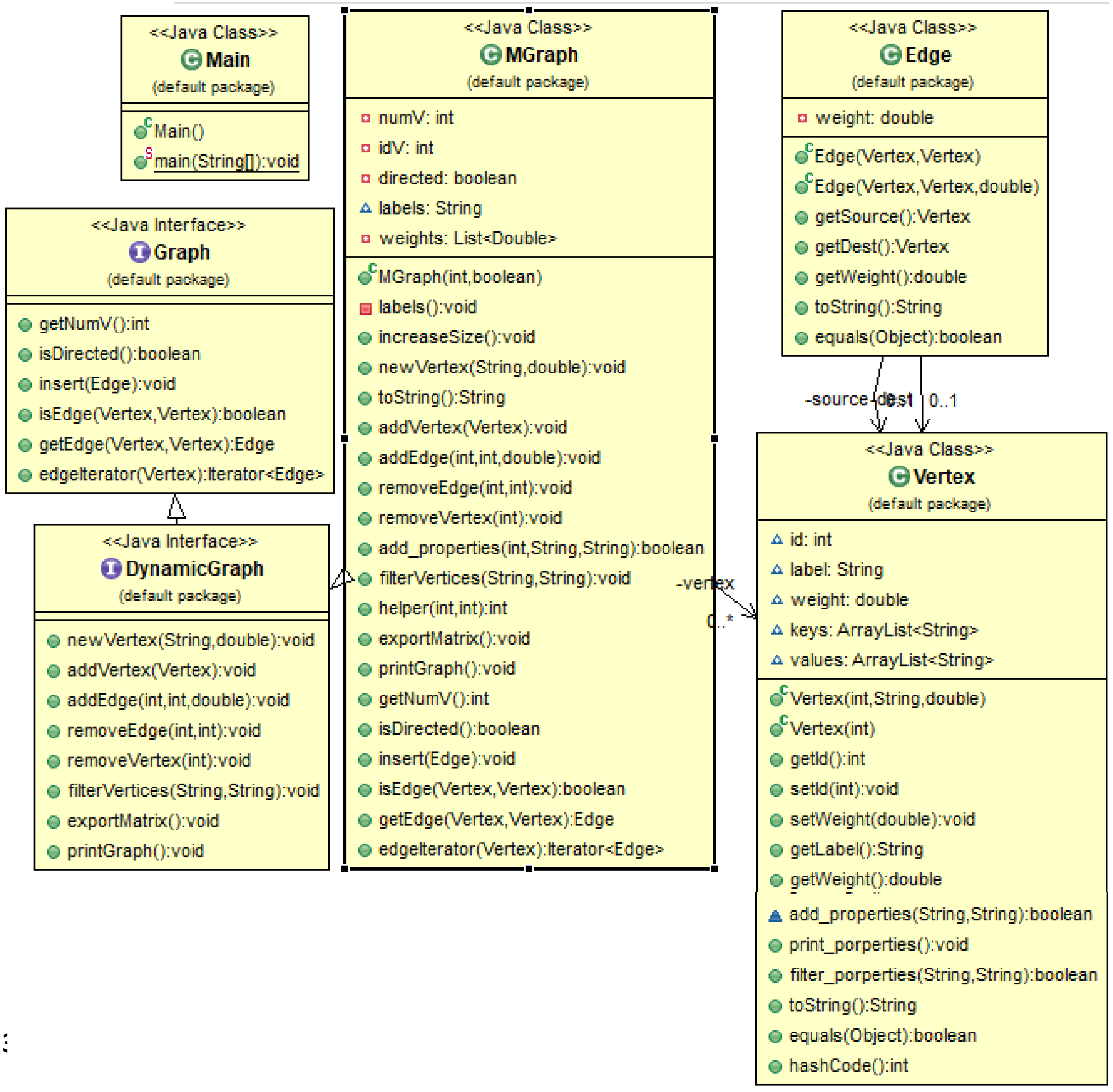
I created a list for the weights of the edges in my MyGraph class.



```java
/** An array of Lists to contain the edges that
 originate with each vertex. */
private List< Vertex >[] vertex;
/**
 *  An array of Lists to contain the weights that
 *      originate with each edge
 */
private List< Double >[] weights;
```

All functions will be tried when the code is run. It is shown in the Output section and Problem Solution Approach.

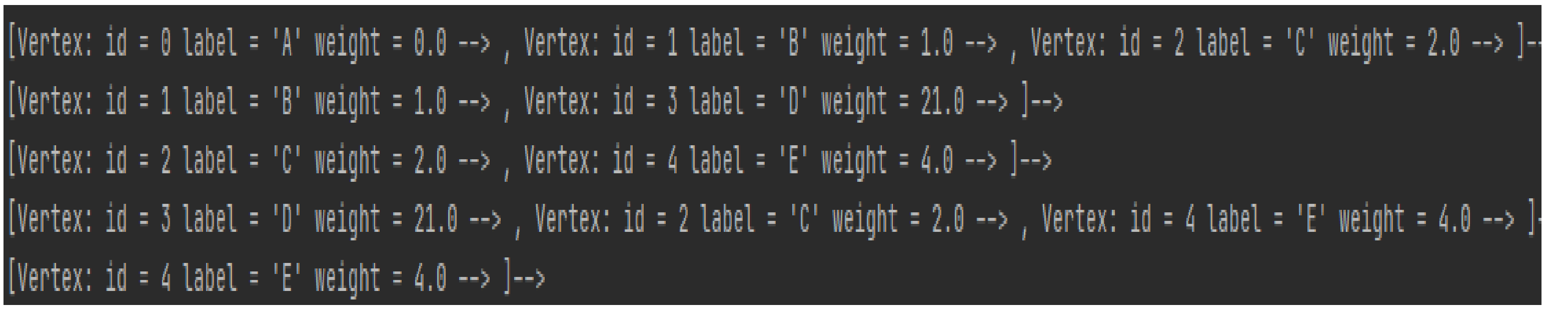
2. USE CASE AND CLASS DIAGRAMS



4. PROBLEM SOLUTION APPRAOACH

The Main Problem was keeping the graph structure with vertex objects without using edge class. The Main Problem was keeping the graph structure with vertex objects without using edge class. In the array I created for this, first the indexes give their vertex numbers. And in each array, it shows the vertex they are connected to with linklists. As shown in the figure, each vertex is connected to other vertexes to which it is connected by edges.

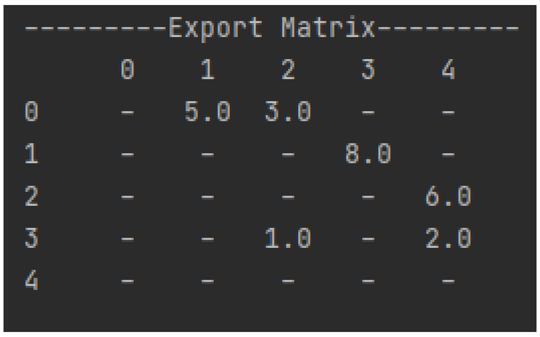
adjacency list :



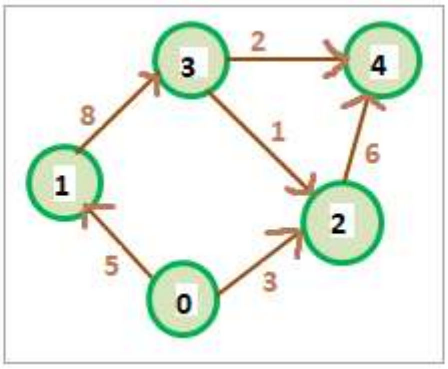
```
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]-
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

The problem is that I couldn't hold the weight of the edges with the vertex class. So I kept a list array of weights inside the Mgraph class. At the point where each edge is added, I saved the edge weights in the weight list. So I could suppress the weights in the matrix representation.

Output as matrix representation of the graph:



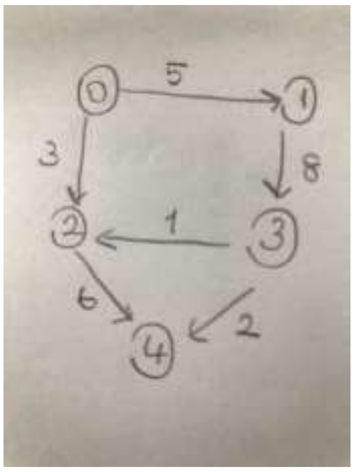
```
---------Export Matrix---------
        0    1    2    3    4
0       -   5.0  3.0   -    -
1       -    -    -   8.0   -
2       -    -    -    -   6.0
3       -    -   1.0   -   2.0
4       -    -    -    -    -
```

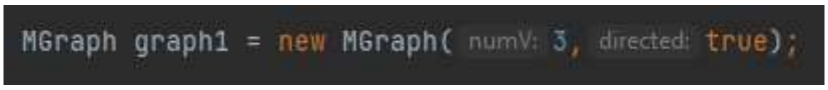


5 TEST CASES AND RUNNING AND RESULTS

Directed

We will create the graph as follows. Then we start experimenting with functions.



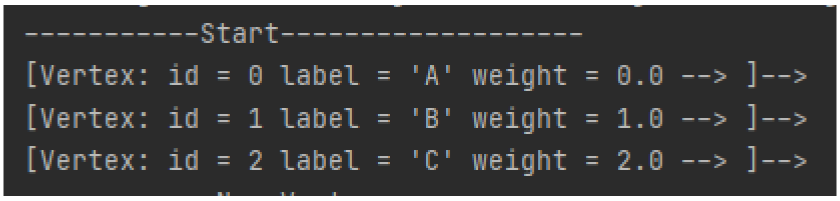


```
MGraph graph1 = new MGraph( numV: 3, directed: true);
```

Object was created with vertex with ids 0, 1, 2.

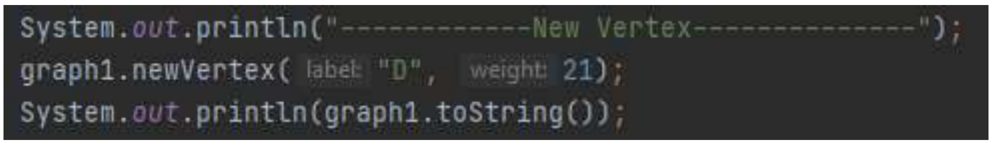
Vertexes 3 and 4 were added by other methods.

- printGraph():



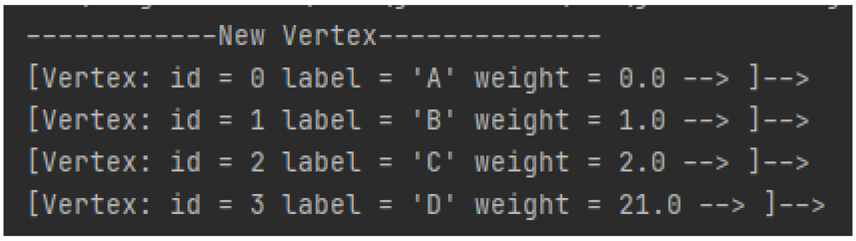
```
-----------Start-------------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
```

- newVertex (string label, double weight):



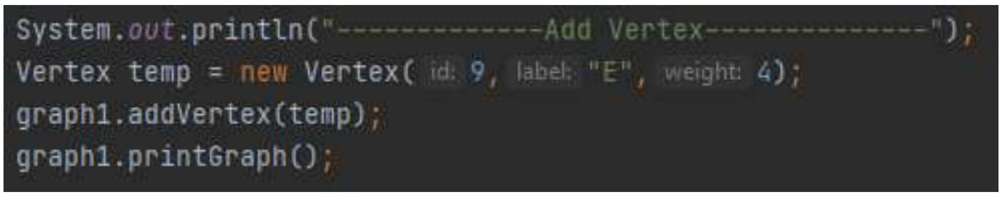
```
System.out.println("------------New Vertex--------------");
graph1.newVertex( label: "D", weight: 21);
System.out.println(graph1.toString());
```

Output:



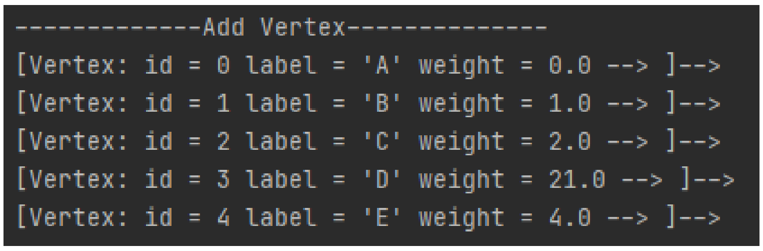
```
------------New Vertex--------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
```

- addVertex (Vertex new_vertex):



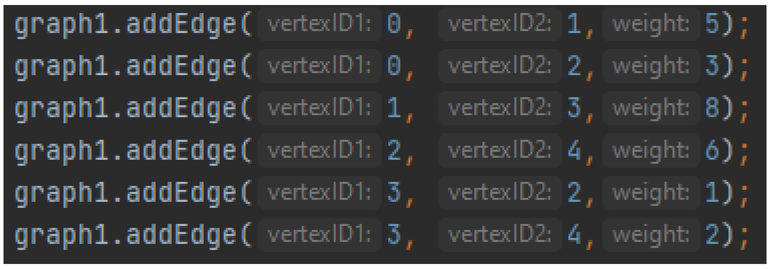
```
System.out.println("-------------Add Vertex--------------");
Vertex temp = new Vertex( id: 9, label: "E", weight: 4);
graph1.addVertex(temp);
graph1.printGraph();
```

Output:



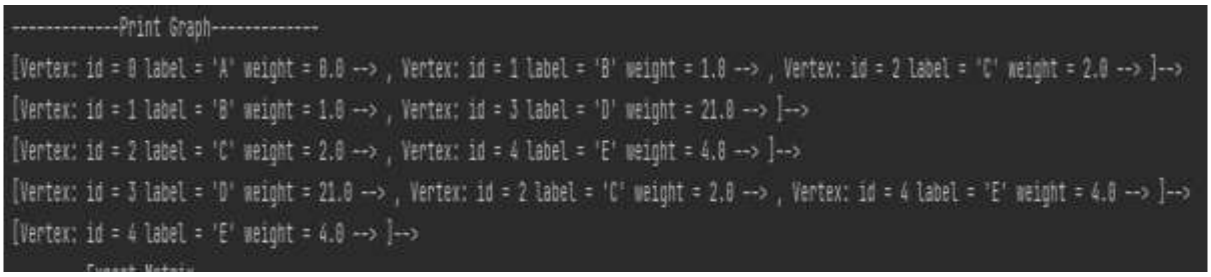
```
-------------Add Vertex--------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

- addEdge (int vertexID1, int vertexID2, double weight):



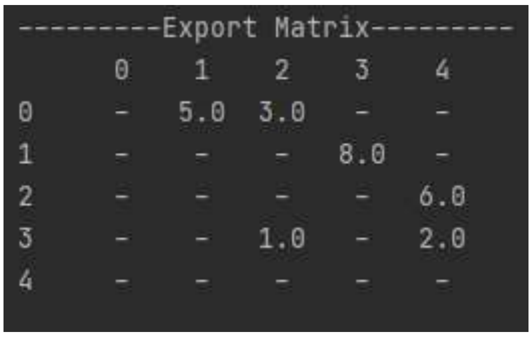
```
graph1.addEdge( vertexID1: 0,  vertexID2: 1, weight: 5);
graph1.addEdge( vertexID1: 0,  vertexID2: 2, weight: 3);
graph1.addEdge( vertexID1: 1,  vertexID2: 3, weight: 8);
graph1.addEdge( vertexID1: 2,  vertexID2: 4, weight: 6);
graph1.addEdge( vertexID1: 3,  vertexID2: 2, weight: 1);
graph1.addEdge( vertexID1: 3,  vertexID2: 4, weight: 2);
```

Output:



```
------------Print Graph------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

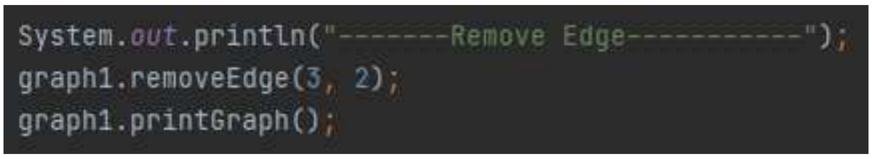
- exportMatrix():



```
---------Export Matrix---------
      0    1    2    3    4
0     -    5.0  3.0  -    -
1     -    -    -    8.0  -
2     -    -    -    -    6.0
3     -    -    1.0  -    2.0
4     -    -    -    -    -
```

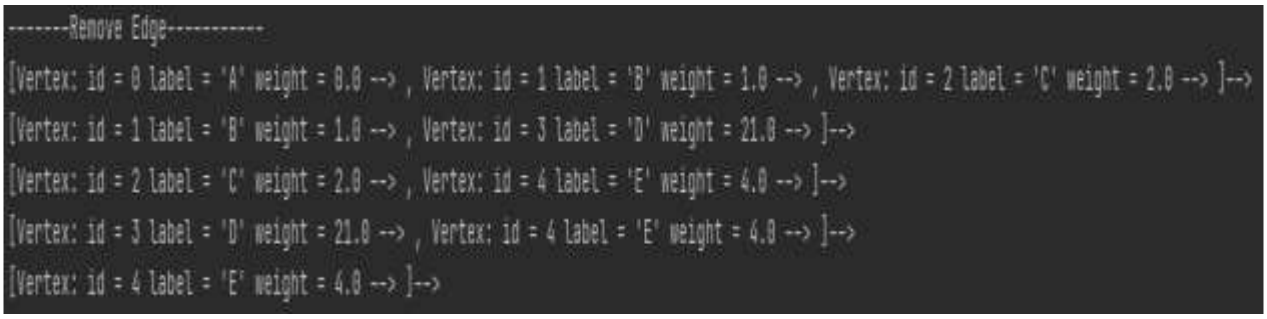
It has printed the matrix display of the above graph to the screen.

- removeEdge (int vertexID1, int vertexID2):



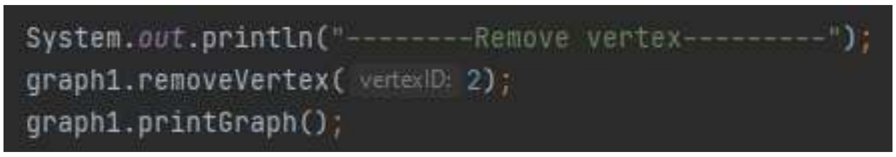
```
System.out.println("-------Remove Edge----------");
graph1.removeEdge(3, 2);
graph1.printGraph();
```

Output:



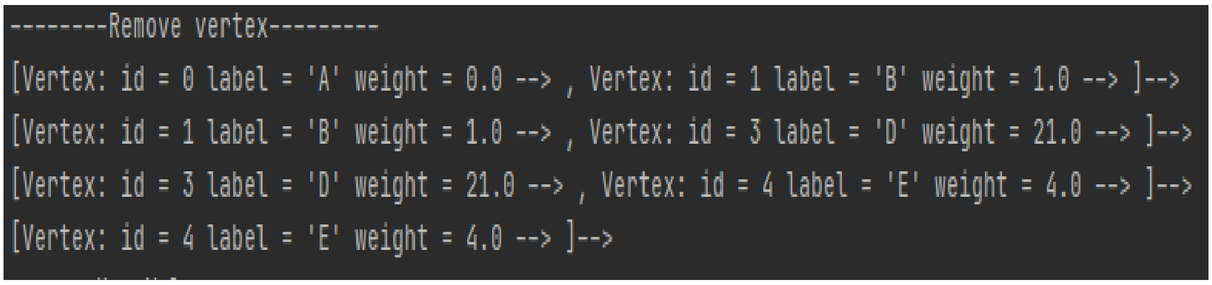
```
-------Remove Edge----------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

- removeVertex (int vertexID) :



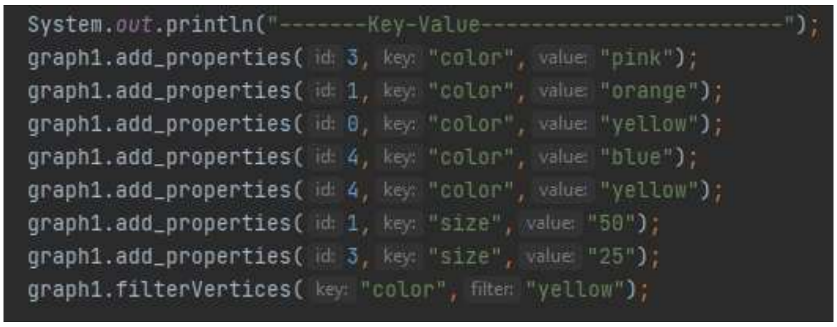
```
System.out.println("--------Remove vertex---------");
graph1.removeVertex( vertexID: 2);
graph1.printGraph();
```

Output:



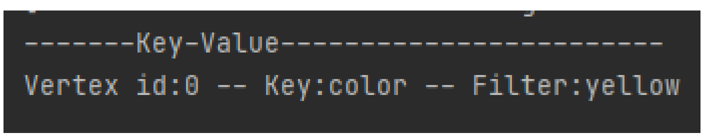
```
--------Remove vertex---------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

- filterVertices (string key, string filter):



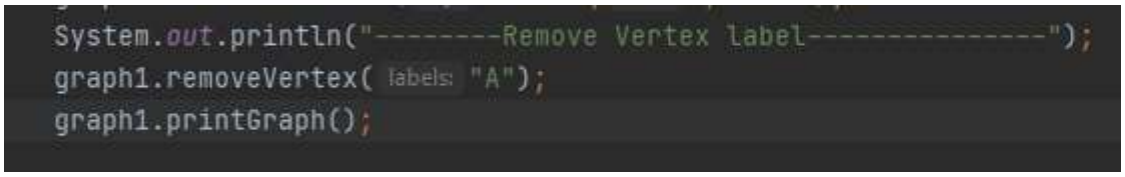
```
System.out.println("-------Key-Value-----------------------");
graph1.add_properties( id: 3, key: "color", value: "pink");
graph1.add_properties( id: 1, key: "color", value: "orange");
graph1.add_properties( id: 0, key: "color", value: "yellow");
graph1.add_properties( id: 4, key: "color", value: "blue");
graph1.add_properties( id: 4, key: "color", value: "yellow");
graph1.add_properties( id: 1, key: "size", value: "50");
graph1.add_properties( id: 3, key: "size", value: "25");
graph1.filterVertices( key: "color", filter: "yellow");
```

Output:



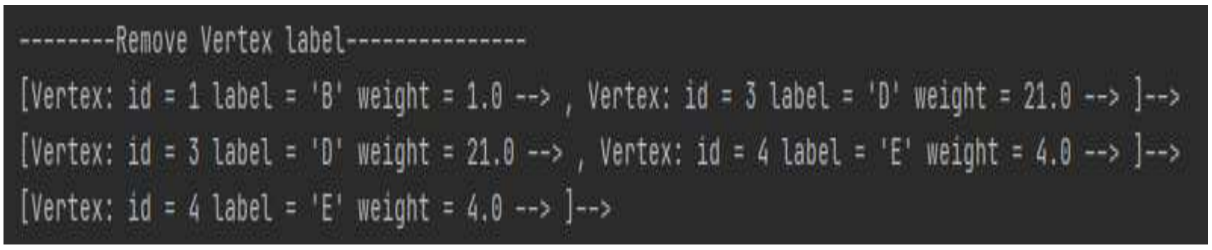
```
-------Key-Value------------------------
Vertex id:0 -- Key:color -- Filter:yellow
```

- removeVertex (string label):



```
System.out.println("--------Remove Vertex label---------------");
graph1.removeVertex( labels: "A");
graph1.printGraph();
```

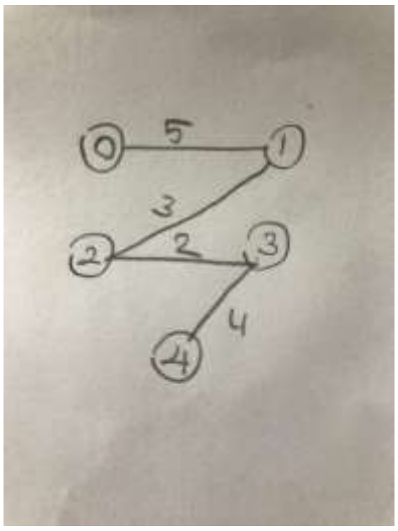
Output:



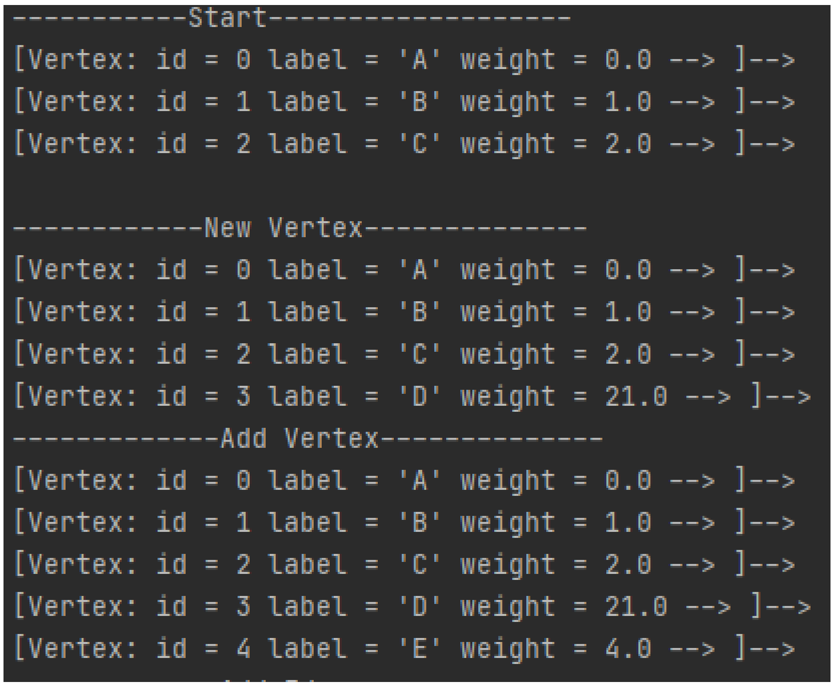
```
--------Remove Vertex label---------------
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```

!isDirected

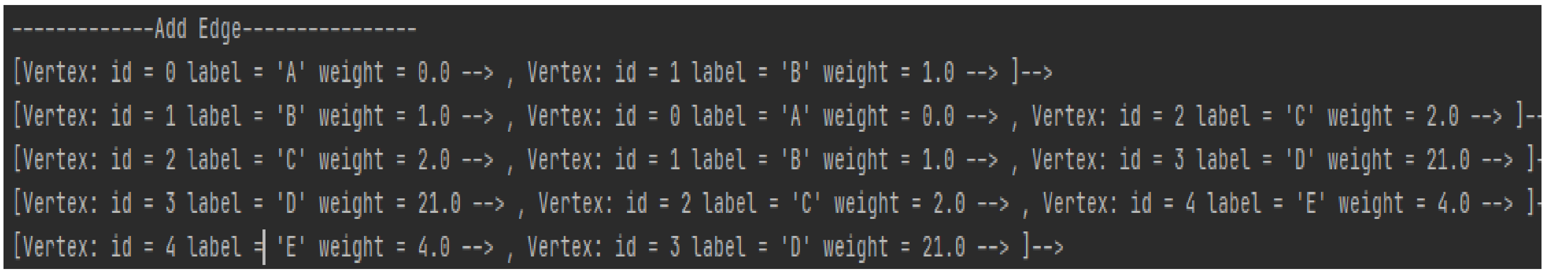
We will create the graph as follows. Then we start experimenting with functions.



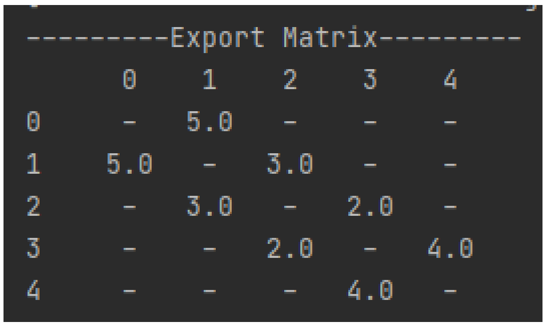
All the methods applied for the directional graph have been applied.



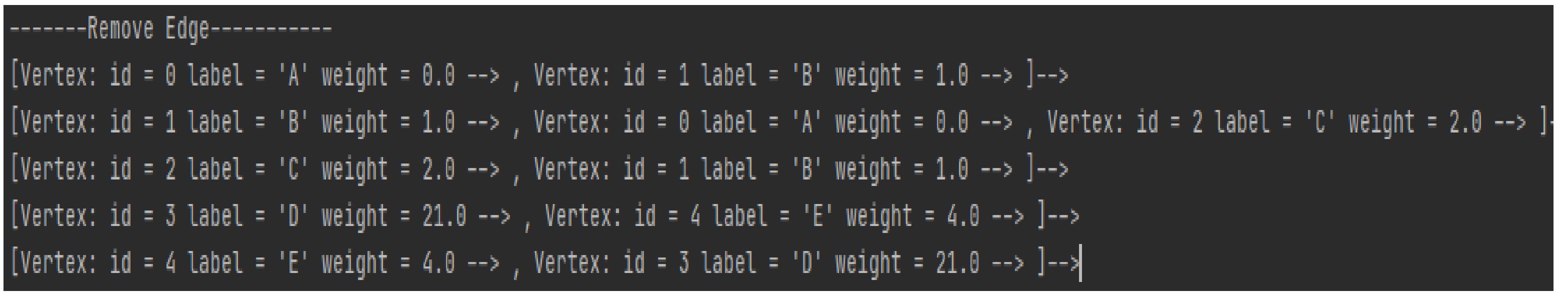
```
------------Start--------------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->

------------New Vertex--------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
-------------Add Vertex--------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 2 label = 'C' weight = 2.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
```



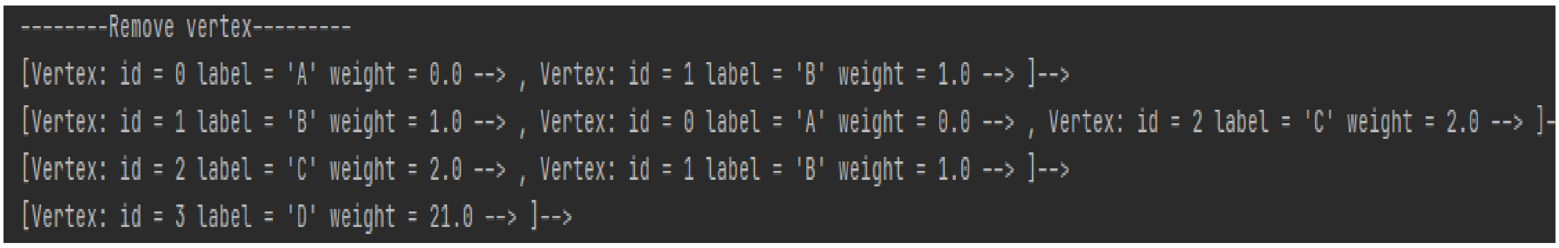
```
-------------Add Edge----------------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]--
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]
[Vertex: id = 4 label = 'E' weight = 4.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
```



```
---------Export Matrix---------
       0    1    2    3    4
0      -   5.0   -    -    -
1     5.0   -   3.0   -    -
2      -   3.0   -   2.0   -
3      -    -   2.0   -   4.0
4      -    -    -   4.0   -
```



```
-------Remove Edge-----------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> , Vertex: id = 4 label = 'E' weight = 4.0 --> ]-->
[Vertex: id = 4 label = 'E' weight = 4.0 --> , Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
```



```
--------Remove vertex---------
[Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 1 label = 'B' weight = 1.0 --> , Vertex: id = 0 label = 'A' weight = 0.0 --> , Vertex: id = 2 label = 'C' weight = 2.0 --> ]
[Vertex: id = 2 label = 'C' weight = 2.0 --> , Vertex: id = 1 label = 'B' weight = 1.0 --> ]-->
[Vertex: id = 3 label = 'D' weight = 21.0 --> ]-->
```