Adjacent List

Default Code:

```
AdjList list = new AdjList();
list.addVertex("A");
list.addVertex("B");
list.addVertex("C");
list.addVertex("D");
list.addVertex("E");
list.addVertex("F");
list.addVertex("G");
list.addVertex("H");

list.addEdge("A", "B", 1);
list.addEdge("B", "C", 2);
list.addEdge("D", "B", 3);
list.addEdge("C", "E", 5);
list.addEdge("E", "F", 6);
list.addEdge("C", "G", 7);
list.addEdge("G", "E", 7);
list.addEdge("G", "H", 4);

list.updateWeightEdge("G", "H", 5);
list.updateWeightEdge("A", "B", 10);
list.updateWeightEdge("D", "B", 9);

list.outNearestNeighbours(-1, "B");
list.inNearestNeighbours(-1, "F");
list.outNearestNeighbours(-1, "C");
list.outNearestNeighbours(1, "C");

list.getEdgeWeight("A", "B");
list.getEdgeWeight("D", "A");

list.removeVertex("J");
list.removeVertex("A");

list.updateWeightEdge("B", "C", 0);
list.updateWeightEdge("E", "F", 0);
list.updateWeightEdge("F", "E", 0);


list.printVertices(null);
list.printEdges(null);
```

Scenario 1

Removing vertex code:

```
long startTime = System.nanoTime();

list.removeVertex("A");

list.removeVertex("B");

list.removeVertex("C");

list.removeVertex("D");

long totalTime = System.nanoTime() - startTime;

double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;

System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Removing edges code:

```
long startTime = System.nanoTime();
list.updateWeightEdge("G", "H", 5);
list.updateWeightEdge("A", "B", 10);
list.updateWeightEdge("D", "B", 9);
list.updateWeightEdge("E", "F", 9);
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Scenario 2

In-neighbours code:

```
long startTime = System.nanoTime();
list.inNearestNeighbours(-1, "F");
list.inNearestNeighbours(-1, "C");
list.inNearestNeighbours(2, "B");
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds");
```

Out-neighbours code:

```
long startTime = System.nanoTime();
list.outNearestNeighbours(-1, "B");
list.outNearestNeighbours(-1, "C");
list.outNearestNeighbours(1, "C");
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds");
```

Scenario 3

Changing edge weight code:

```java
long startTime = System.nanoTime();
list.updateWeightEdge("G", "E", 5);
list.updateWeightEdge("B", "C", 3);
list.updateWeightEdge("E", "F", 9);
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds");
```

Incidence Matrix

Default Code:

```java
IncidenceMatrix matrix = new IncidenceMatrix();
matrix.addVertex("A");
matrix.addVertex("B");
matrix.addVertex("C");
matrix.addVertex("D");
matrix.addVertex("E");
matrix.addVertex("F");
matrix.addVertex("G");
matrix.addVertex("H");

matrix.addEdge("A", "B", 1);
matrix.addEdge("B", "C", 2);
matrix.addEdge("D", "B", 3);
matrix.addEdge("C", "E", 5);
matrix.addEdge("E", "F", 6);
matrix.addEdge("C", "G", 7);
matrix.addEdge("G", "E", 7);
matrix.addEdge("G", "H", 4);

matrix.updateWeightEdge("G", "H", 5);
matrix.updateWeightEdge("A", "B", 10);
matrix.updateWeightEdge("D", "B", 9);

matrix.outNearestNeighbours(-1, "B");
matrix.inNearestNeighbours(-1, "F");
matrix.outNearestNeighbours(-1, "C");
matrix.outNearestNeighbours(1, "C");

matrix.getEdgeWeight("A", "B");
matrix.getEdgeWeight("D", "A");

matrix.removeVertex("J");
matrix.removeVertex("A");

matrix.updateWeightEdge("B", "C", 0);
matrix.updateWeightEdge("E", "F", 0);
matrix.updateWeightEdge("F", "E", 0);


matrix.printVertices(null);
matrix.printEdges(null);
```

Scenario 1

Removing Vertex code:

```
long startTime = System.nanoTime();

matrix.addVertex("A");

matrix.addVertex("B");

matrix.addVertex("C");

matrix.removeVertex("A");

long totalTime = System.nanoTime() - startTime;

double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;

System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Removing Edges code:

```
long startTime = System.nanoTime();
matrix.addVertex("A");
matrix.addVertex("B");
matrix.addVertex("C");
matrix.addVertex("D");
matrix.addEdge("A", "B", 1);
matrix.updateWeightEdge("A", "B", 0);
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Scenario 2

In-neighbours code:

```
long startTime = System.nanoTime();
matrix.addVertex("A");
matrix.addVertex("B");
matrix.addVertex("C");
matrix.addVertex("D");
matrix.addEdge("A", "B", 1);
matrix.inNearestNeighbours(1, "B");
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Out-neighbors code:

```java
long startTime = System.nanoTime();
matrix.addVertex("A");
matrix.addVertex("B");
matrix.addVertex("C");
matrix.addVertex("D");
matrix.addEdge("A", "B", 1);
matrix.outNearestNeighbours(1, "B");
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```

Scenario 3

Changing edge weight code:

```java
long startTime = System.nanoTime();
matrix.addVertex("A");
matrix.addVertex("B");
matrix.addEdge("A", "B", 1);
matrix.updateWeightEdge("A", "B", 3);
long totalTime = System.nanoTime() - startTime;
double elapsedTimeInSeconds = (double) totalTime / 1_000_000_000;
System.out.println("Total Time: " + elapsedTimeInSeconds + " seconds.");
```