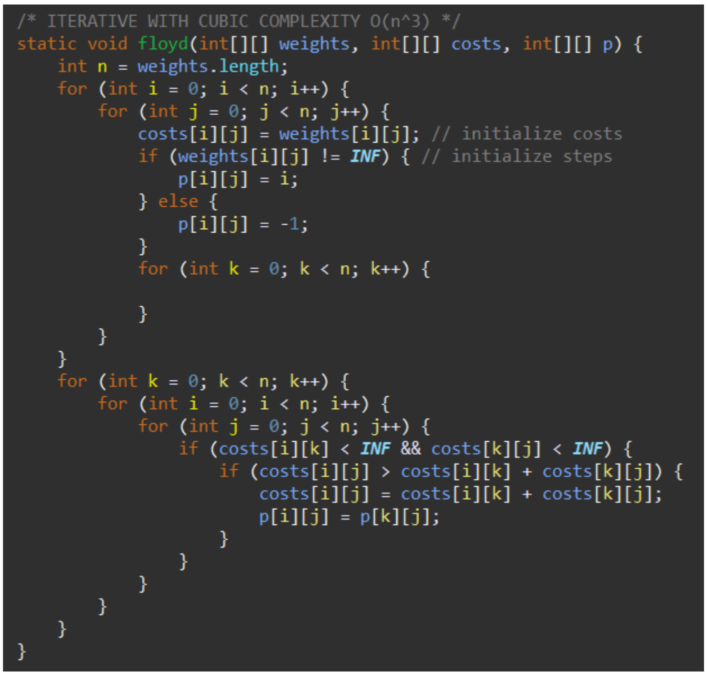
Algorithmics	Student information	Date	Number of session
	UO: 296503	24/03/2025	Session5
	Surname: Mulet Alonso		
	Name: Sergio		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Minimum path

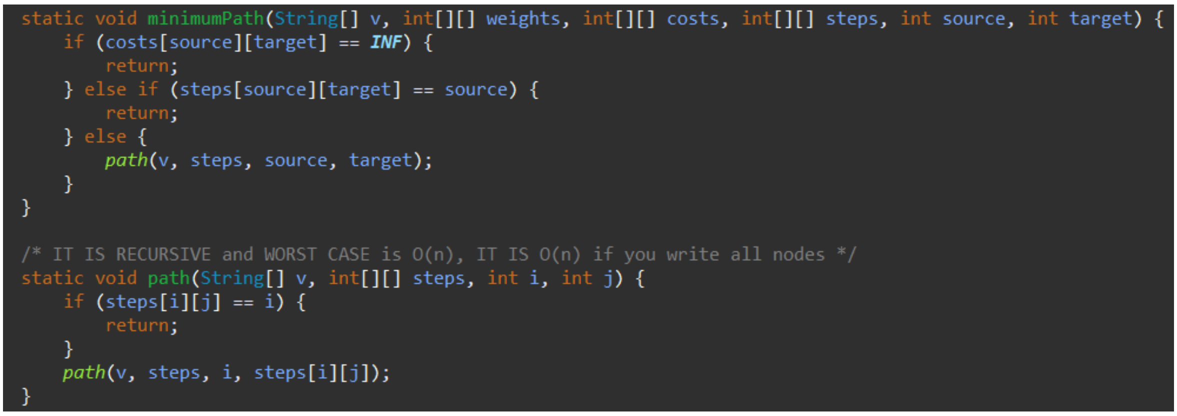
MinimumPathsExample:

First, I completed Floyd method, it has a complexity of O(n^3)



```java
/* ITERATIVE WITH CUBIC COMPLEXITY O(n^3) */
static void floyd(int[][] weights, int[][] costs, int[][] p) {
    int n = weights.length;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            costs[i][j] = weights[i][j]; // initialize costs
            if (weights[i][j] != INF) { // initialize steps
                p[i][j] = i;
            } else {
                p[i][j] = -1;
            }
            for (int k = 0; k < n; k++) {

            }
        }
    }
    for (int k = 0; k < n; k++) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (costs[i][k] < INF && costs[k][j] < INF) {
                    if (costs[i][j] > costs[i][k] + costs[k][j]) {
                        costs[i][j] = costs[i][k] + costs[k][j];
                        p[i][j] = p[k][j];
                    }
                }
            }
        }
    }
}
```

Then minimumPath() and path():



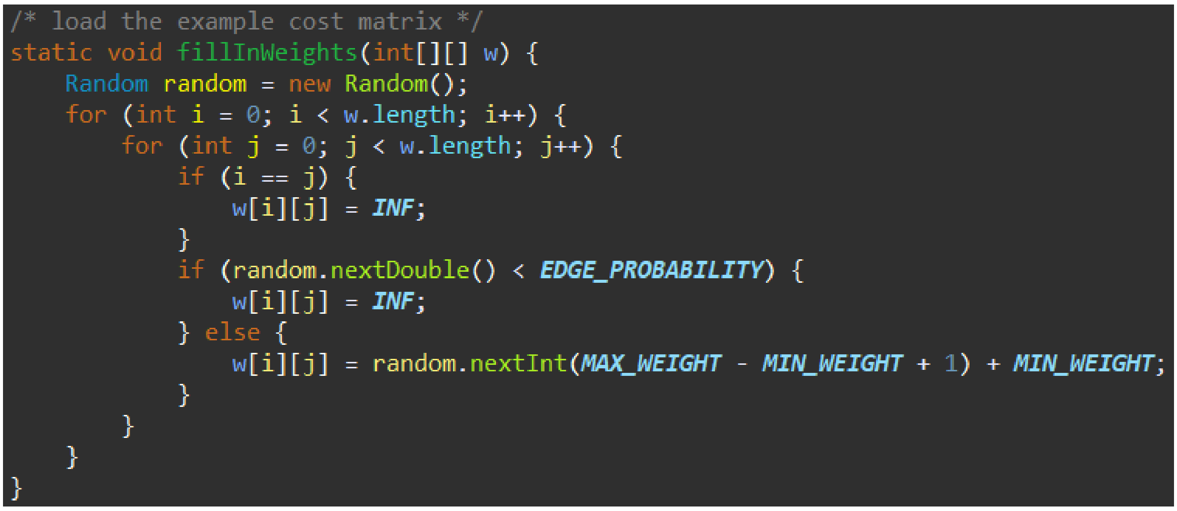
```java
static void minimumPath(String[] v, int[][] weights, int[][] costs, int[][] steps, int source, int target) {
    if (costs[source][target] == INF) {
        return;
    } else if (steps[source][target] == source) {
        return;
    } else {
        path(v, steps, source, target);
    }
}

/* IT IS RECURSIVE and WORST CASE is O(n), IT IS O(n) if you write all nodes */
static void path(String[] v, int[][] steps, int i, int j) {
    if (steps[i][j] == i) {
        return;
    }
    path(v, steps, i, steps[i][j]);
}
```

Then, by using the methods provided in class we can se the output.

	Student information	Date	Number of session
Algorithmics	UO: 296503	24/03/2025	Session5
	Surname: Mulet Alonso		
	Name: Sergio		

MinimumPaths:

It is basically the same class but generating the matrices randomly, here is the method I have implemented.

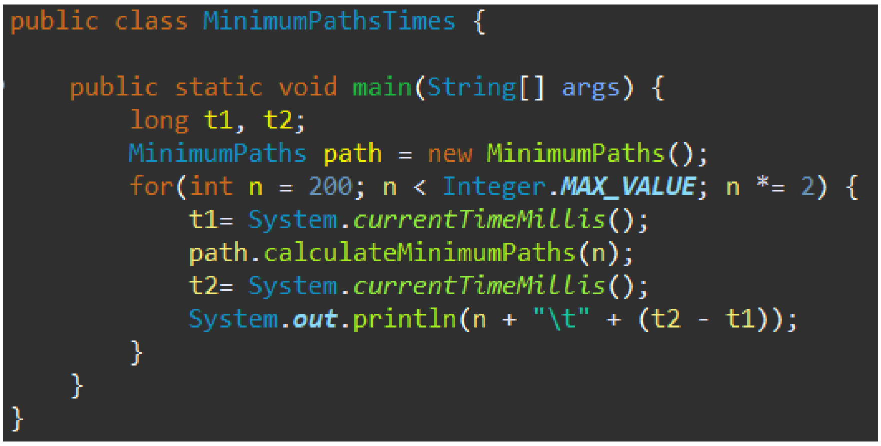


```java
/* load the example cost matrix */
static void fillInWeights(int[][] w) {
    Random random = new Random();
    for (int i = 0; i < w.length; i++) {
        for (int j = 0; j < w.length; j++) {
            if (i == j) {
                w[i][j] = INF;
            }
            if (random.nextDouble() < EDGE_PROBABILITY) {
                w[i][j] = INF;
            } else {
                w[i][j] = random.nextInt(MAX_WEIGHT - MIN_WEIGHT + 1) + MIN_WEIGHT;
            }
        }
    }
}
```

Where MAX_WEIGHT = 99, MIN_WEIGHT = 10, INF = 100000 and EDGE_PROBABILITY = 0,5

MinimumPathsTimes:

I have created a simple class for measuring times passing to MinimumPath the size of the problem, which goes from 200 to integer max value.



```java
public class MinimumPathsTimes {

    public static void main(String[] args) {
        long t1, t2;
        MinimumPaths path = new MinimumPaths();
        for(int n = 200; n < Integer.MAX_VALUE; n *= 2) {
            t1= System.currentTimeMillis();
            path.calculateMinimumPaths(n);
            t2= System.currentTimeMillis();
            System.out.println(n + "\t" + (t2 - t1));
        }
    }
}
```

Algorithmics	Student information	Date	Number of session
	UO: 296503	24/03/2025	Session5
	Surname: Mulet Alonso		
	Name: Sergio		

Times obtained:

n	t(ms)
200	318
400	2488
800	19196
1600	147033
3200	OoT

In this case, OoT means that it takes more than 5 minutes.

Floyd algorithm has a complexity of O(n^3), which matches the times I got executing the algorithm. It is more or less clear to see that by increasing the problem size by 3, the times are being multiplied by 3^3 (9).

Theoretically, for 3200 nodes it will take about 22 minutes to compute the minimum paths.