To run this program, you need open the file WPFI_Algorithm.java, then run the main function inside it

If you want to modify minSup, threshold, alpha. You should modify at here

property	value
Minsup ratio	Double (0 <= msup_ratio <= 1)
threshold	Double (0 <= threshold <= 1)
Alpha (scale factor)	Double (0 < scale_factor <= 1)

```
// modify input parameter
double threshold = 0.7;
double alpha = 0.6;

// modify list of ratio for minimum support parameter
double[] ratios = new double[]{0.1, 0.15, 0.2, 0.25, 0.3};

try{
    database.loadUnData(path: "data/connect.txt");
```

If you want to change the dataset, firstly you need to put the dataset file in the same root with WPFI_Algorithm.java and then change the parameter of the method loadData and replace parameter with the name of dataset you want test. Extension of file name dataset is required.

```
try{
    database.loadData(path: "connect.txt");
```

If you want to try the fully database, you can use the method converFullyDB to create a fully database and use method changefully to convert partly to fully database. If you not use this feature, by default the algorithm will use partly database

```
try{
    database.loadData(path: "connect.txt");
    database.convertFullyDB();
    database.changeFully();
```

If you want to test all the dataset, you should follow this step:

```
Run | Debug
public static void main(String[] args) {
    UncertainDatabase database = new UncertainDatabase(lines: 10000);
```

Remove the lines parameter when Initialize the UncertainDatabase if you want read all the content of dataset. Otherwise, set the number of lines to take the first n lines of the dataset

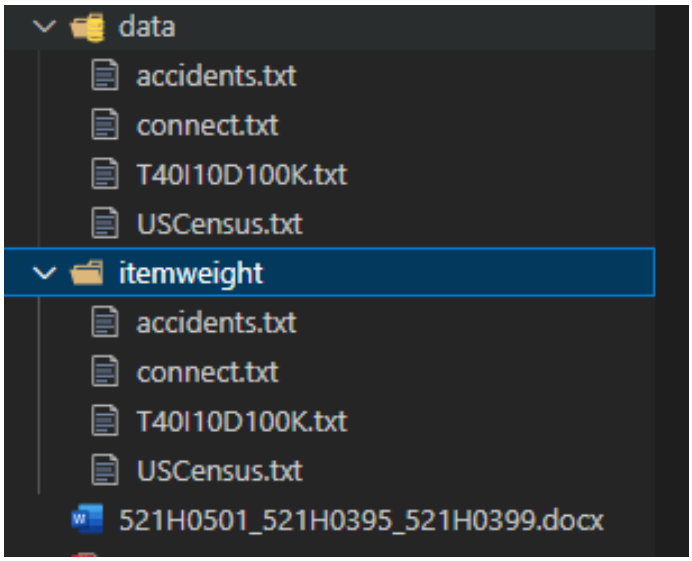
If you want change the algorithm 2 to algorithm 3. You will set the parameter useProbabilityModel of runAlgorithm method to true in case you want to use algorithm 3 and set to false in case you want to use algorithm 2.

```
hm 2
os) {
atio * database.size());
tem.nanoTime();
 apriori.runAlgorithm(database, minSup, threshold, alpha, useProbabilityModel: false);
m.nanoTime();
 (endTime - startTime) / 1000000000;
took: "
    + executionTime + "s");
```

This is the version of code used to benchmark algorithm 2 and algorithm 3.

In the root directory, there are 2 subdirectories: data and itemweight.

- The ‘data’ directory: used to store the uncertain database after loading data from the deterministic database.
- The ‘itemweight’ directory: used to store the weight table as described in the report."



If the ‘data’ or ‘itemweight’ folder is empty, you have to use the loadData method to create a new uncertain database

```
try{
    database.loadData(path: "connect.txt");
```

After using loadData method, 2 new files will be created in ‘data’ and ‘itemweight’ folder and they have the same name with the deterministic database which you have read before

When you want to reuse the uncertain database that was created earlier, please use the following method:

```
try{
    database.loadUnData(path: "data/connect.txt");
```

NOTE:

_ If both the ‘data’ and ‘itemweight’ directories already have an uncertain database (with the same name as the deterministic database), when reusing the ‘loadData’ method, it will create a new uncertain database with a new probability value and overwrite the old file

_ This version only applies to partly uncertain database.

_ The fully uncertain database still working but it will be created a new fully uncertain database when you recompile the code (use loadData method to apply the fully uncertain database)

By default, the data and itemweight directories will contain the data that I use to benchmark algorithm 2 and algorithm 3 in the report