

## Project p1\_40354020\_172

The purpose of this project is to find the intersection between an array of sets using four different strategies. The project is divided into two parts. The first consists of writing an algorithm for the implementation of each strategy and testing it to make sure that all the strategies find the intersection correctly. It will take as input the strategy to be executed (no input to execute all strategies) and the program will output to the console a set defined by the intersection of all the sets acquired from the files located in the “inputFiles” folder. The second part of the project is designed to compare the execution time of all four strategies. The program will take as input certain parameters from which it will produce many data sets of increasing sizes and will output a file in the “experimentalResults” folder with a table containing the time each strategy took to find the intersection for each data set.

For this specific test, the data to be generated is based on the scenario that there are a number of crime events for which each telephone company in the area has provided a file containing the telephone numbers associated with each crime event. Each set generated has all the numbers from all the companies associated for a given crime event. The goal is to find the intersection to see which numbers appear in all events.

### Prerequisites

The code was compiled using JavaSE-9 so you must have java updated.

### Instructions to execute Part1Main from the Command Prompt

To execute Part1Main from the terminal, you must first go to the project directory. Once in the directory execute the following command:

**java -classpath ./bin p1MainClasses/Part1Main *n***      where  $1 \leq n \leq 4$

*n* = 1 is for P1 strategy, *n* = 2 is for P2 strategy, *n* = 3 is for P3 strategy and *n* = 4 is for P4 strategy.

To execute all strategies simultaneously simply execute the following command:

**java -classpath ./bin p1MainClasses/Part1Main**

Example 1:

-Command: `java -classpath ./bin p1MainClasses/Part1Main 2`

-Output: Final set by P2: {7, 2, 1, 3, 4};

Example 2:

-Command: `java -classpath ./bin p1MainClasses/Part1Main`

-Output: Final set by P1: {7, 2, 1, 3, 4};

Final set by P2: {1, 2, 3, 4, 7};

Final set by P3: {1, 2, 3, 4, 7};

Final set by P4: {1, 2, 3, 4, 7};

## Instructions to execute Part2Main from the Command Prompt

To execute Part2Main from the terminal, you must first go to the project directory. Once in the directory execute the following command:

**java -classpath ./bin p1MainClasses/Part2Main *n m isize fsize istep rep***

integer *n* – number of companies

integer *m* – number of crime events

integer *isize* – initial size of dataset

integer *fsize* – final size of dataset

integer *istep* – rate at which to increment size

integer *rep* – number of repetitions per dataset

To use default values (*n* = 20, *m* = 50, *isize* = 1000, *fsize* = 50000, *istep* = 1000, *rep* = 200) execute the same command without parameters:

**java -classpath ./bin p1MainClasses/Part2Main**

Using any other number of parameters other than six or none will throw an `IllegalArgumentException`.

Example 1:

-Command: **java -classpath ./bin p1MainClasses/Part2Main 20 50 1000 20000 1000 200**

-Output(File allResults.txt):

Size	P1	P2	P3	P4
1000	1104376.2	284072.06	1002473.94	522897.16
2000	292625.66	268241.4	489671.3	565834.7
3000	296997.3	229492.6	444779.84	324659.56
4000	288162.62	194935.92	309621.28	206577.88
5000	370878.38	250954.62	302726.84	235283.05
6000	449313.97	295873.12	342481.38	272426.78
7000	577717.06	389272.47	474074.25	367651.3
8000	582586.4	356375.12	380861.03	339924.75

9000	578434.94	342393.38	341492.6	310295.12
10000	635804.6	374979.2	383834.25	336071.25
11000	719282.3	427513.38	415851.1	375683.53
12000	797653.6	447329.7	450963.1	401666.25
13000	811858.94	457556.16	432406.53	393799.9
14000	1076260.2	601441.06	578252.0	521904.88
15000	1167421.4	646958.94	643254.4	582390.06
16000	1037806.06	540668.1	524624.06	476495.38
17000	1098413.0	597357.2	555777.5	479008.0
18000	1131313.9	580862.9	539977.4	508160.16
19000	1240235.9	653511.4	586206.25	521935.44
20000	1327821.0	675532.06	631091.0	585156.6