Project – 4 : Implementation of Single Flow-Spread Sketches

The following project contains a zip file, which consists of 3 source code files(java) which contains the logic for implementing BitMap, Probabilistic BitMap and HyperLogLog algorithms and corresponding output text files.

Below are the steps which guides to run the project on command line to the functionality of the algorithms mentioned above.
1) Launch the terminal application and go to the "src" folder of the project.
2) Begin by executing the following commands:
"javac P4_ BitMap.java", "java P4_ BitMap 10000" and
"javac P4_ProbabilisticBitMap.java", "java P4_ProbabilisticBitMap 10000 0.1" and
"javac 4_HLL.java", "java P4_HLL 256"
3) Executing these commands will result in the creation of output files in the "src" directory and, if those files already exist, will replace them.

1. **BitMap.java:**
   In this algorithm, we try to evaluate the hash for the each of the element and we perform the hash for it and assign the bitmap value to 1 and estimation of the flow spread is done using the formula "-BitMap_Size * ln(fraction of elements that are zeros)"

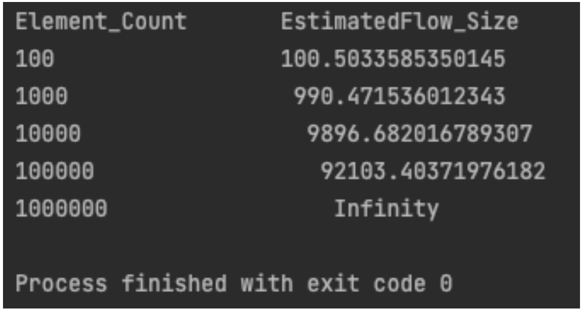
2. **ProbabilisticBitMap.java:**
   In this algorithm, we calculate the max hash that is for the flow and assign the bitmap value of the hash to 1 when we have the Hash value is less than the max hash * sampling probability. This step is repeated for the entire flows and we estimate the flow spread using the formula "-(BitMap_Size/ Probablity) * ln(fraction of elements that are zeros)".

3. **HLL.java:**
   In this algorithm, we have 5 bits for each of the register as it is given the size of 256, we evaluate the hash for each of the element and we calculate the number of leading zeros for the hash value and we use the information to evaluate the flow spread using the formula "Alpha * BitMap_Size^2(1/2^HLLBitMap[Hash(Element)])^-1"

This project is tested out on both IntelliJ and command line for execution, and below are the screenshots for the outputs. Project starts when above commands is run and contains the calls to the methods to execute BM.generateFlowElements, PBM.generateFlowElements and HLL.HLL_Recording.

Output Screenshots:

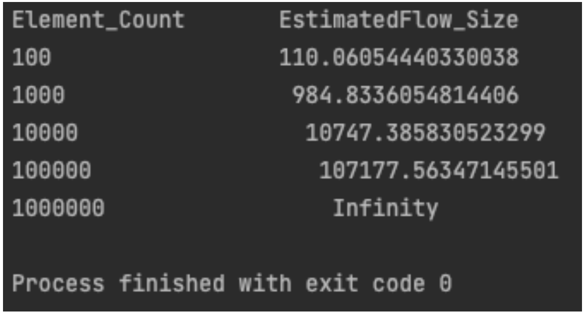
1. **BitMap:**



```
Element_Count       EstimatedFlow_Size
100                 100.5033585350145
1000                 990.471536012343
10000                  9896.682016789307
100000                  92103.40371976182
1000000                  Infinity

Process finished with exit code 0
```

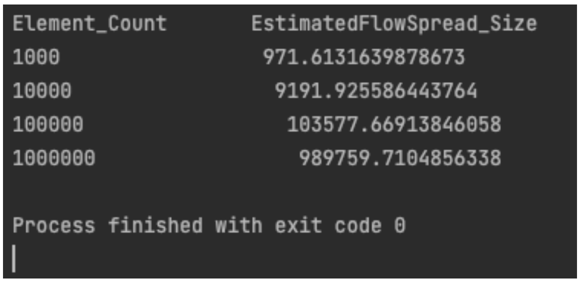
2. ProbabilisticBitMap:



```
Element_Count       EstimatedFlow_Size
100                 110.06054440330038
1000                 984.8336054814406
10000                 10747.385830523299
100000                 107177.56347145501
1000000                 Infinity

Process finished with exit code 0
```

3. HLL:



```
Element_Count       EstimatedFlowSpread_Size
1000                 971.6131639878673
10000                 9191.925586443764
100000                 103577.66913846058
1000000                 989759.7104856338

Process finished with exit code 0
```