**IDS Project 5**

**VirtualBitmap**

Here bitmap is being used to check the spread of each flow in a multi flow spread.

Each flow is assigned a virtual bitmap of certain size which is mapped to physical bitmap we hash its value to an index in the bitmap and set it to 1. Ideally the number of 1’s would be the number of unique elements in the flow, which is nothing but the spread. But there are always chance of hash collisions so we take several parameters into consideration to formulate the flow spreads estimated value.

*Implementation:*

We have used a very simple approach to deal with the bitmap.

* In the class VirtualBitmap we have declared the size m of bitmap globally and the bitmap itself as an array of size m, and an array flowIDs which holds all the flows. We also have our random object to call generate random numbers.
* When we take a flow of given spread size, we invoke the constructor with that size and then create the flow array of that size.
* We make sure that the bitmap is filled with zeroes by using inbuilt method arrays.fill() and set them to zeroes before hashing
* We use a method fillRandomValues() which takes in an array as argument to generate unique positive values.
* For recording we perform two hashes to map.
* For finding estimated spread we use query() which first calculates the number of zeroes in bitmap and virtual bitmap then substitute in the formula discussed in class for the estimated spread value.
* **We have to keep in mind the case of saturated bitmaps where all the bits are set to 1 which means that the number of zeroes will be 0, that is our u=0, there by leading our log value to infinity.**
* **In these cases, our output will be infinity which is the integer.max**
* We store the output of this program in a text file plotpoints.txt
* We use this plot points and feed it as data to out python program and form the scatterplot and saved as OutputGraph.pdf

Throughout our project we used Math.abs to make sure the random numbers generated are made positive only.

**bSketchHLL**

Here bitmap with registers is being used to check the spread of each flow in a multi flow spread.

The bitmap acts as a data structure which holds hyper log log sketches.

When we receive a flow, we hash its flow id to know which row to put it in and then and another hash for column number and thne the perform geometric hashing whose value is the number of leading zeroes in hash value generated and then decide what value will . The recoding is slightly different as we store the bitmap value itself as maximum of uniform hash and geometric hash. But there are always chance of hash collisions so we take several parameters into consideration to formulate the flow spreads estimated value.

*Implementation:*

* In the class BsketchHLL, we have declared the size m of bitmap globally and the bitmap itself is an data structure and an array flow which holds all the elements related to one flow. We also have our random object to call generate random numbers.
* When we take a flow of given spread size, we initialize the constructor with that size and then create the flow array of that size. We fill it with unique random positive values. We generate hash and we make sure that the bitmap is filled with zeroes by using inbuilt method arrays. Fill() and set them to zeroes before hashing.
* We initialize a 2D array of size A[4000][128]
* We use flow ids and random hashes k, to find which estimators to store in and the flow elements to store in which register of HLL.
* To perform recording we first hash the element H(e) and get its value and also find the geometric hash G(e) and record the cell value by encoding bitmap A[row][col] to the maximum{ A[row][col],G(e)}.
* We use a method fill Random Elements() which takes in an array as argument to generate unique positive values.
* We use method G which takes in integer and returns its number of leading zeroes in binary form of given number.
* For querying we use estimated Spread() which first calculates the constant alpha from given formula and then , the harmonic sum of all the given bitmap values and the substitute in the formula discussed in class for the estimated spread value.
* We sort the estimated spread values in descending order and print top 25 values as output.
* **Keep in mind that bSketch Hyper Log log value is not accurate when the flow spread is small.**
* We store the output of this program in a text file OutputBSketchHLL.txt

For ease of flow, we have clubbed several operations but we can also split the methods for recording and querying.