

NETWORK DATA STREAMING  
VASU NEGI  
UFID: 8495-3933

### PROJECT-3

In order to run these programs on \*nix based/MAC system, follow the instructions below:

- \* Go to project directory in terminal
- \* Type the command in the following way:

```
python3 file_name
```

```
#####CountMin#####
```

Run: python3 CountMin.py

For Demo, following values are taken , n = 10,000, k = 3, w = 3000

Functions:

1. def query(self,key):  
Returns the minimum size of the flow in all the counters
2. def recording(self,key):  
This function records the size of each flow in counters
3. def generate\_random\_numbers(self):  
generate random numbers
4. def get\_hash\_functions(self, number):  
generate hashfunctions with the following function - > (number XOR i) where i is any random number in the self.random\_nums
5. def execute(self):  
Record the sizes of all flows in the counter,query to evaluate estimated sizes of all flows, query for all flows, and calculate the average error of all flows.  
Also, Print the (flowid, estimated size, true size) of the 100 largest estimated size

Output: Output/CountMin\_output.txt  
The output files are in the Output folder.

1. 1st line of the output: the average error among all flows
2. Next 100 lines: (flowid, estimated size, true size) of the largest estimated size

#### #####Counter Sketch#####

Run: python3 CounterSketch.py

For demo: following values are taken:  $n = 10,000$ ,  $k = 3$ ,  $w = 3000$

Functions:

1. def query(self,key):  
Returns the median size of the flow in all the counters
2. def recording(self,key):  
This function will record the size of the flow (flowid) in the counter, depending on the MSB of the hash\_value. If the MSB is 1, we add, else we subtract the packet.
3. def generate\_random\_numbers(self):  
generate random numbers
4. def get\_hash\_functions(self, number):  
generate hashfunctions with the following function -  $\rightarrow (\text{number XOR } i)$  where  $i$  is any random number in the self.random\_nums
5. def execute(self):  
Record the sizes of all flows in the counter,query to evaluate estimated sizes of all flows, query for all flows, and calculate the average error of all flows.  
Also, Print the (flowid, estimated size, true size) of the 100 largest estimated size

Output: Output/CounterSketch\_output.txt

The output files are in the Output folder.

1. 1st line of the output: the average error among all flows
2. Next 100 lines: (flowid, estimated size, true size) of the largest estimated size

#### #####ActiveCounter#####

Run: python3 ActiveCounter.py

For demo, counter will run 1,000,000 times, and number part is of 16 bits and exponent is of 16 bits.

Functions:

1. def increment(self):  
This function will increment the counter. The counter is only incremented based on the probability calculated based on the exp part.  
If the number  $n$  overflows, then right shift  $n$  and increment  $e$

2. def probability(self):

This function implements the probability of incrementing the counter, as  $(1 / 2^{c.e})$

Output: Output/ActiveCounter\_output.txt

The output files are in the Output folder.

1. The final value of the active counter in decimal