Asmt 4: Frequent Items

1 Streaming Algorithms

A (20 points): Run the Misra-Gries Algorithm (see **L11.3.1**) with $(k - 1) = 9$ counters on streams S1 and S2. Report the output of the counters at the end of the stream.

In S1:

Label	j	c	a	o	b	v	e	z	n
Counter	1	104715	194715	1	147715	1	1	0	1

In S2:

Label	e	b	g	c	a	m	w	t	k
Counter	1	121429	1	161430	231429	0	0	0	0

In each stream, use just the counters to report how many objects *might* occur more than 20% of the time, and which must occur more than 20% of the time.

In S1:
3 objects might occur more than 20% of the time
0 object must occur more than 20% of the time

In S2:
3 objects might occur more than 20% of the time
'a' must occur more than 20% of the time

B (20 points): Build a Count-Min Sketch (see **L12.1.1**) with $k = 10$ counters using $t = 5$ hash functions. Run it on streams S1 and S2.

For both streams, report the estimated counts for objects a, b, and c.

In S1.txt
Estimate count for a = 266737
Estimate count for b = 203000
Estimate count for c = 160000

In S2.txt
Estimate count for a = 309544
Estimate count for b = 184971
Estimate count for c = 210000

Just from the output of the sketch, which of these objects, with probably $1-\delta = 31/32$ (that is assuming the randomness in the algorithm does not do something bad), *might* occur more than 20% of the time?

In S1:
a might occur more than 20% of the time: True
b might occur more than 20% of the time: True
c might occur more than 20% of the time: False

In S2:
a might occur more than 20% of the time: True
b might occur more than 20% of the time: False
c might occur more than 20% of the time: True

C (5 points): How would your implementation of these algorithms need to change (to answer the same questions) if each object of the stream was a "word" seen on Twitter, and the stream contained all tweets concatenated together?

If we are interested in the "word" object in the stream of concatenated tweets, the current algorithms might take in account concatenated words or stop words. We can remove the stop words and separate possible concatenated words in the preprocessing stream step, or the algorithms must recognize these words. In Misra-Gries, instead of checking if the stream element matches a label, the algorithm needs to check if a substring of the stream element partially or fully matches a label, and the stream element should not be a stop word. It is tricky to do the similar thing with Count-Min Sketch algorithm, because Count-Min Sketch doesn't save the label. Therefore, we must preprocess the stream before using Count-Min Sketch.

D (5 points): Describe one advantage of the Count-Min Sketch over the Misra-Gries Algorithm.

Count-Min Sketch guarantees that q is a heavy hitter if count(q) = $\widehat{f_q}$ is large enough, but it might not true for Misra-Gries