# Problem 1

*Suppose that we want to use Count-Min Sketch to select heavy hitters in a stream. All items that are $\frac{1}{k}$ - heavy hitters should be selected. In addition, any item that is not a $\frac{1}{2k}$ - heavy hitter should have only 0.1% chance of being selected.*

- *Explain how this can be done if the length of the stream m is known in advance.*
- *How would you modify your algorithm if m is not known in advance?*

*Remark: Note that Count-Min Sketch itself does not store any elements. It only maintains the counts. The goal here is to identify and store all the heavy hitters.*

Run Misra-Gries with $k-1$ counters in parallel with CMS. This will give us the candidate $\frac{1}{k}$ - heavy hitters. CMS should be run with $\delta = 0.001$, $\epsilon = \frac{1}{2k}$, $l = log\left(\frac{1}{\delta}\right) = log(1000)$, and $b = \frac{2}{\epsilon} = 4k$. The estimates obtained with CMS will be in the range $f_x \leq \hat{f}_x \leq f_x + \frac{m}{2k}$ with probability 99.9%.

Finally, output the elements produced by Misra-Gries for which the CMS frequency estimate is $\geq \frac{m}{k}$. The resulting set of elements contains all $\frac{1}{k}$ - heavy hitters, and for each element in this set the probability that it isn't at least a $\frac{1}{2k}$ - heavy hitter is 0.1%. No prior knowledge of $m$ is required.

# Problem 2

*In the Count-Min Sketch algorithm we discussed in class, when we encounter an element x, we increment all counters associated with that element. Suppose that we only increment the subset of these counters that currently have the minimum value among the counters associated with x. Does this still work? Is this a good idea?*

**Intuition:**

Consider the following scenario:

The first element in the stream will increment all of its counters, since all counters are set to zero.

The second element comes with two cases: either it is the same as the first element, or it is distinct. The first case is not interesting, however in the second case there are three options:

1. counters of the $2^{nd}$ element are distinct from the counters of the $1^{st}$ element.
   - not interesting - all counters of $2^{nd}$ element are incremented.
2. all counters of the $2^{nd}$ element are the same as the counters of the $1^{st}$ element.
   - extremely unlikely, and not interesting - all counters of $1^{st}/2^{nd}$ element are incremented.

3. some counters of the $2^{nd}$ element are distinct from the counters of the $1^{st}$ element and some are overlapping.

In the third scenario, only the counters unique to $2^{st}$ element will be incremented, since those counters will be still at 0, whereas the counters overlapping with the $1^{st}$ element will have already been incremented previously.

**Observation:**.

*If we consider each increment of a counter as an event, the total number of increments is at most equal to that of classical CMS and is likely to be lower. Since each time an element is seen at least one of its counters will be incremented and counter will only be omitted if they are already bigger, it is clear that an underestimate is impossible.*

There exists another scenario, in which an element could have no unique counters, but share them with several other elements of the stream (likely the most common scenario for large streams of many distinct elements). In this case, the final count is likely to be an overestimate. We can observe, however, that since the total number of increments is at most as many as in classical CMS and is likely lower, the extent of the overestimate is likely lower.

**Conclusion:**

Lower bound on the produced estimate is at least the actual true count, and the upper bound will be lower, so this sounds like a good idea.