L3S Research Center
Large Scale Data Mining, SS 2016
Dr. Avishek Anand
Solution to Assignment 2, due: **28 April 2016**

Problem 1. Suppose we have a stream of tuples with the schema:

Grades(university, courseID, studentID, grade)

Assume universities are unique, but a courseID is unique only within a university (i.e., different universities may have different courses with the same ID, e.g., CS101) and likewise, studentID's are unique only within a university (different universities may assign the same ID to different students). Suppose we want to answer certain queries approximately from a 1/20th sample of the data. For each of the queries below, indicate how you would construct the sample. That is, tell what the key attributes should be.

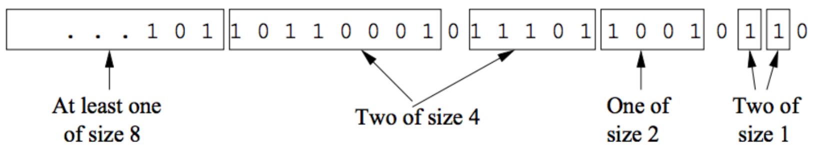
1. For each university, estimate the average number of students in a course.
2. Estimate the fraction of students who have a GPA of 3.5 or more.
3. Estimate the fraction of courses where at least half the students got 'A'.

Solution:

1. For each university, select course ID as the key attribute.
2. Use a combination of university ID + student ID to avoid getting duplicates.
3. Use a combination of course ID + university ID.

Problem 2. Consider the DGIM algorithm for counting 1's in a stream. Suppose the window is as shown in the figure. Estimate the number of 1's the the last k positions, for k = (a) 5 (b) 15. In each case, how far off the correct value is your estimate?

```
. . 1 0 1 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 0
```

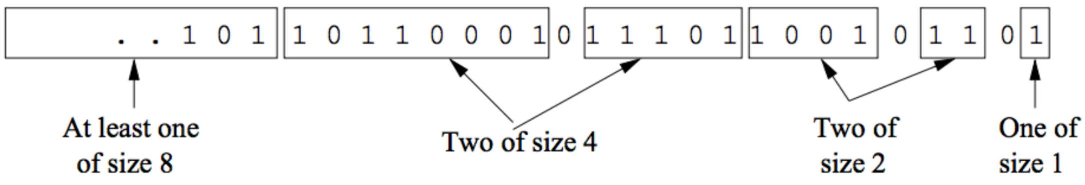


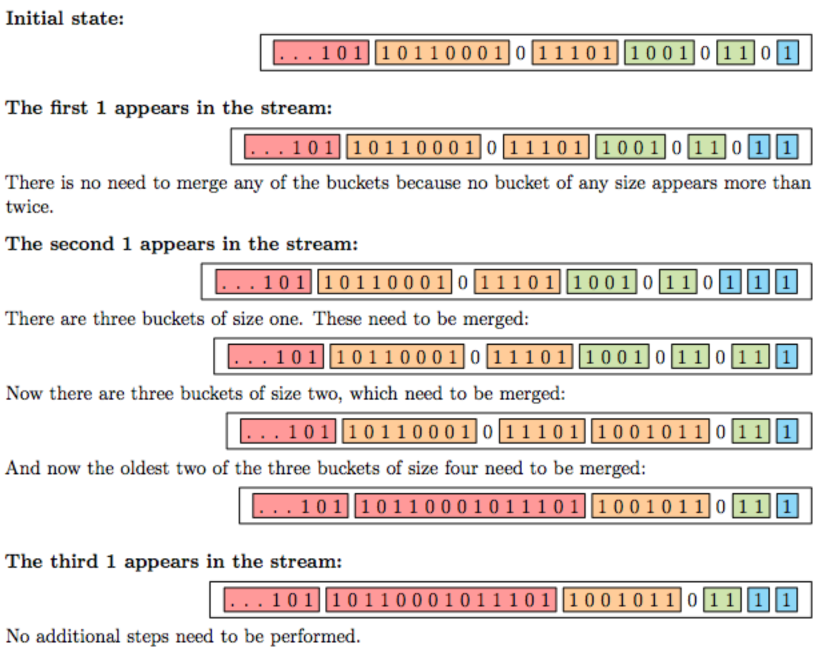
Solution: To estimate the number of $1's$, first add the bucket sizes until the last bucket before the k^{th} bit. Divide the size of the bucket that holds the k^{th} bit by 2 and add it to the sum.

1. $k = 5$. Estimate is 3 and the correct value is also 3.
2. $k = 15$. Estimate is 10 and the correct value is 9.

Problem 3. Describe what happens to the buckets if three more 1's enter the window represented by the figure below. You may assume none of the 1's shown leave the window.

L3S Research Center
Large Scale Data Mining, SS 2016
Dr. Avishek Anand
Solution to Assignment 2, due: **28 April 2016**




Initial state:

The first 1 appears in the stream:

There is no need to merge any of the buckets because no bucket of any size appears more than twice.

The second 1 appears in the stream:

There are three buckets of size one. These need to be merged:

Now there are three buckets of size two, which need to be merged:

And now the oldest two of the three buckets of size four need to be merged:

The third 1 appears in the stream:

No additional steps need to be performed.

Solution:

Problem 4. Now let us consider an algorithm which uses fixed window sizes to count 1's (say window width w), then what is the:

- estimated space requirement for storing windows and their representations?
- maximum error you can expect when querying for the last k bits where $k \leq N$?

Solution:

- Space required = Number of buckets * Space needed for representing bucket size

  Number of buckets $= \frac{N}{w}$

  Space to represent bucket of size $w = log_2{}^w$

  Space required $= \frac{N}{w} log_2{}^w$

- To compute the maximum error we look at 2 scenarios: first where k is large and second when k is very small.
  The first observation to note in such a scenario is that the rate of error decreases linearly as k increases. Since the bucket size is fixed, the number of 1's being counted correctly

increases linearly as more buckets are encountered. The error will only occur in the bucket containing the k-th bit. At worst the error in this bucket is w.
In the second case where k is small, i.e. k lies in the first bucket, the error is unbounded since we don't know how many 1's each bucket is supposed to contain. We are only given the width of the bucket. The bucket could contain as many as w 1's or even zero 1's.