08: Data Streams

Number of questions:	5
Positive points per question:	3.0
Negative points per question:	1.0

Gradiance quiz on Data Streams. You can attempt to answer the questions as many times as you like. Questions get randomly regenerated each time. The score of the *last* submission gets saved into our records (that is, once you get a perfect score, don't submit again with a bad one).

1. A certain Web mail service (like gmail, e.g.) has 10^8 users, and wishes to create a sample of data about these users, occupying 10^{10} bytes. Activity at the service can be viewed as a stream of elements, each of which is an email. The element contains the ID of the sender, which must be one of the 10^8 users of the service, and other information, e.g., the recipient(s), and contents of the message. The plan is to pick a subset of the users and collect in the 10^{10} bytes records of length 100 bytes about every email sent by the users in the selected set (and nothing about other users).

The method of Section 4.2.4 will be used. User ID's will be hashed to a bucket number, from 0 to 999,999. At all times, there will be a threshold t such that the 100-byte records for all the users whose ID's hash to t or less will be retained, and other users' records will not be retained. You may assume that each user generates emails at exactly the same rate as other users. As a function of n, the number of emails in the stream so far, what should the threshold t be in order that the selected records will not exceed the 10^{10} bytes available to store records? From the list below, identify the true statement about a value of n and its value of t.

- a) $n = 10^{10}$; $t = 10{,}000$
- b) $n = 10^{11}$; $t = 1000$
- c) $n = 10^{11}$; $t = 999$
- d) $n = 10^9$; $t = 999$

2. Suppose we are using the DGIM algorithm of Section 4.6.2 to estimate the number of 1's in suffixes of a sliding window of length 40. The current timestamp is 100, and we have the following buckets stored:

End Time	100	98	95	92	87	80	65
Size	1	1	2	2	4	8	8

Note: we are showing timestamps as absolute values, rather than modulo the window size, as DGIM would do.

Suppose the query asks us to estimate the number of 1's in the 6 most recent elements. What are the largest and smallest number of 1's that could be the correct answer, given the information above? What would the DGIM algorithm estimate? Identify the true statement in the list below.

- a) The minimum possible number of 1's is 5.

b) The maximum possible number of 1's is 5.

c) The estimated number of 1's is 1.

d) The maximum possible number of 1's is 3.

3. Here is an example of a tiny Bloom filter. It uses an array of 10 bits and two independent hash functions f and g. We want to test membership in a set S of three elements, so we hash each of the three elements using both f and g, and we set to 1 any bit that any of the three elements is hashed to by either of the hash functions.

When a new element x arrives, we compute f(x) and g(x), and we say x is in the set S if both f(x) and g(x) are 1. Assume x is not in the set S. What is the probability of a false positive; i.e., the probability of saying that x is in S. Identify this probability, correct to three decimal places, in the list below.

a) .780 b) .220 c) .121 d) .178

4. We wish to use the Flagolet-Martin lgorithm of Section 4.4 to count the number of distinct elements in a stream. Suppose that there ten possible elements, 1, 2,..., 10, that could appear in the stream, but only four of them have actually appeared. To make our estimate of the count of distinct elements, we hash each element to a 4-bit binary number. The element x is hashed to $3x + 7$ (modulo 11). For example, element 8 hashes to 3*8+7 = 31, which is 9 modulo 11 (i.e., the remainder of 31/11 is 9). Thus, the 4-bit string for element 8 is 1001.

A set of four of the elements 1 through 10 could give an estimate that is exact (if the estimate is 4), or too high, or too low. You should figure out under what circumstances a set of four elements falls into each of those categories. Then, identify in the list below the set of four elements that gives the exactly correct estimate.

a) { 2, 6, 8, 10}

b) {4, 5, 6, 7}

c) {4, 5, 6, 10}

d) {1, 5, 8, 9}

5. Suppose we hash the elements of a set S having 12 members, to a bit array of length 101. The array is initially all-0's, and we set a bit to 1 whenever a member of S hashes to it. The hash function is random and uniform in its distribution. What is the expected fraction of 0's in the array after hashing? What is the expected fraction of 1's? You may assume that 101 is large enough that asymptotic limits are reached.

a) The fraction of 1's is $1-e^{-12/101}$.

b) The fraction of 1's is $e^{-89/101}$.

c) The fraction of 0's is 12/101.

d) The fraction of 1's is $1-e^{-89/101}$.