# Part A. Mining Data Streams

1. DGIM Algorithm

Bucket Information:

|  |  |  |
| --- | --- | --- |
| Bucket id | Size | Timestamp |
| 1 | 1 | 1000 |
| 2 | 2 | 997 |
| 3 | 2 | 992 |
| 4 | 4 | 990 |
| 5 | 8 | 984 |
| 6 | 8 | 969 |
| 7 | 16 | 955 |
| 8 | 32 | 917 |
| 9 | 64 | 856 |
| 10 | 64 | 729 |
| 11 | 128 | 597 |
| 12 | 256 | 341 |

We have 12 buckets in total, and the final estimate is **457**.

1. The Flajolet-Martin Algorithm

* For h(x) = 3x + 1 mod 32, the binary result is [00100, 11100, 01010, 10000, 00100, 10011, 10000, 00100, 01101], tail length = 4, estimate result is 16
* For h(x) = 2x + 7 mod 32, the binary result is [01001, 11001, 01101, 10001, 01001, 10011, 10001, 01010, 01111], tail length = 0, estimate result is 1
* For h(x) = 8x mod 32, the binary result is [01000, 01000, 11000, 01000, 01000, 10000, 01000, 01000, 00000], tail length = 4 (5 if 00000 is considered as 5 rather than 1), estimate result is 16 (32 if you regard the tail length of 00000 as 5).

The problem of the second and third hash functions:

1. The right most bit of the **second** hash function will always be **1**, so the result will always be 1.
2. The right **most three bits** of the **third** hash function will always be **zero**! The final result will always 8

Suggestions:

If someone wants to use a hash function of the form h(x) = ax + b mod 2k, then choosing a from **odd** numbers, b from **prime** numbers is a good choice.