Weekplan: Streaming I

Philip Bille   Inge Li Gørtz   Eva Rotenberg

References and Reading

[1] Amit Chakrabarti: *Data Stream Algorithms* 2011 (updated July 2020) chapter 0 except 0.3 and chapter 1.

We recommend reading the specified chapters and sections of [1] in detail.

Exercises

The following exercise relates to the streaming model. Remember that we use the number of bits when we calculate space in the streaming model.

1 Largest numbers Given m numbers from the universe $[n] = \{1, \ldots, n\}$, suppose we want to find the k largest.

1.1 In the RAM-model, how would you solve this task? What is your total running time?

1.2 In the streaming model, would you solve this task? What is your space usage? What is your running time?

2 Missing numbers

2.1 Assume you get $n-1$ different integers from the set $\{1, \ldots, n\}$ in a stream. Can you deduce the missing number using only $O(\log n)$ space?

2.2 [$*$] Assume now you only get $n-2$ different integers from the set. Can you find the two missing numbers in $O(\log n)$ space?

3 Reservoir sampling[1] Reservoir sampling is a method for choosing an item item uniformly at random from an arbitrarily long stream of data; for example, the sequence of packets that pass through a router, or the sequence of IP addresses that access a given web page. Like all data stream algorithms, this algorithm must process each item in the stream quickly, using very little memory.

Algorithm 1: GETONESAMPLE(stream S)

```
ℓ ← 0
while S is not done do
    x ← next item in S
    ℓ ← ℓ + 1
    if RANDOM(ℓ) = 1 then
        sample ← x          (⋆)
end
return sample
```

Here RANDOM(a) is a random number generator that uniformly at random returns an integer between 1 and a (both included). At the end of the algorithm, the variable ℓ stores the length of the input stream S; this number is not known to the algorithm in advance. If S is empty, the output of the algorithm is (correctly!) undefined. In the following, consider an arbitrary non-empty input stream S, and let n denote the (unknown) length of S.

3.1 Prove that the item returned by GETONESAMPLE(S) is is chosen uniformly at random from S.

[1] This exercise is from Jeff Erickson's notes on streaming

3.2 What is the expected number of times that GETONESAMPLE(S) executes line ($\star$)?

3.3 What is the expected value of ℓ when GETONESAMPLE(S) executes line ($\star$) for the *last* time?

3.4 What is the expected value of ℓ when either GETONESAMPLE(S) executes line ($\star$) for the *second* time or the algorithm ends (whichever happens first)?

3.5 Describe and analyze an algorithm that returns a subset of k distinct items chosen uniformly at random from a data stream of length at least k. The integer k is given as part of the input to your algorithm. Prove that your algorithm is correct.

For example, if $k = 2$ and the stream contains the sequence $\langle \spadesuit, \heartsuit, \diamondsuit, \clubsuit \rangle$, the algorithm should return the subset $\{\diamondsuit, \spadesuit\}$ with probability $1/6$.

The following exercises relate to chapter 1 in [1].

4 Frequency $[w]$ Consider the trivial solution to the frequency problem: Keeping as many counters as there are colours. What is the space-consumption?

5 Misra-Gries $[w]$ Run Misra-Gries' algorithm on the following stream with $k = 3$. What do you output? How large was your largest counter?

b a b b a m b a m b a n a n a n a n a

6 Tightness of Misra-Gries Given k and m, design a stream of length m that contains some character m/k times yet this character is not output by Misra-Gries' algorithm.

7 Exercise 1-1 from [1] Let $\hat{m}$ be the sum of all counters maintained by the Misra-Gries algorithm after it has processed an input stream, i.e., $\hat{m} = \sum_{\ell \in \text{keys}(A)} A[\ell]$. Prove that

$$f_j - \frac{m - \hat{m}}{k} \leq \hat{f}_j \leq f_j \, .$$

8 Merging two streams This exercise is almost identical to exercise 1-3 from [1].

Suppose we have run the (one-pass) Misra-Gries algorithm on two streams σ_1 and σ_2, thereby obtaining a summary for each stream consisting of $k - 1$ counters. Let m_i denote the length of the stream σ_i. Consider the following algorithm for merging these two summaries to produce a single $k - 1$-counter summary.

Algorithm 2: Merge two streams

```
Combine the two sets of counters, adding up counts for any common items.
if more than k − 1 counters remain then
    c ← value of kth counter, based on decreasing order of value.
    Reduce each counter by c.
    Delete all keys with non-positive counters.
end
```

Prove that the resulting summary is good for the combined stream $\sigma_1 \circ \sigma_2$ in the sense that frequency estimates from it satisfy the bounds given for Misra-Gries, namely

$$f_j - \frac{m}{k} \leq \hat{f}_j \leq f_j \, ,$$

where $m = m_1 + m_2$ is the length of the combined stream.