■ **Exercise 1 (Streaming algorithm for frequent items).** We want to design a streaming algorithm that finds all the items in a stream of n items with frequency strictly greater than n/k for some fixed k. Consider the following algorithm:

Algorithm 1 Misra-Gries algorithm for frequent items

```
Initialize: A := empty dictionnary
for i = 1..n do
    if x_i ∈ keys(A) then
        A[x_i] := A[x_i] + 1
    else
        if # keys(A) < k − 1 then
            A[x_i] := 1
        else
            for each a ∈ keys(A) do
                A[a] := A[a] − 1
                if A[a] == 0 then
                    Remove a from A
Output: On query a, if a ∈ keys(A), then report f̂_a := A[a], else report f̂_a := 0.
```

We denote by $f_a = \#\{i : x_i = a\}$ the frequency of a in the stream.

▶ ***Question 1.1)*** *Show that for all* a, $\quad f_a - \dfrac{n}{k} \leqslant \hat{f}_a \leqslant f_a.$

▷ Hint. *Show that the decrement loop is performed at most* $\frac{n}{k}$ *times while reading the stream.*

▶ ***Question 1.2)*** *Conclude that one can find the items with frequency larger than* n/k *with two passes on the stream.*

▶ ***Question 1.3)*** *Let* $\hat{n} = \sum_{a \in \mathsf{keys}(A)} A[a]$*. Show that for all* a, $\quad f_a - \dfrac{n - \hat{n}}{k} \leqslant \hat{f}_a \leqslant f_a.$

■ **Exercise 2 (Streaming algorithm for counting triangles).** We want to estimate the number of triangles in a graph given as a stream of its edges. Let us consider the following algorithm (we assume that the number of vertices and edges, n and m resp., are known).

Algorithm 2 Counting triangles

```
Pick an edge uv uniformly at random in the stream
Pick a vertex w ∈ [n] ∖ {u, v} at uniformly at random
if edges uw and vw appear after edge uv in the stream then
    output m(n − 2)
else
    output 0
```

▶ ***Question 2.1)*** *Show that* $\mathbb{E}[\text{output}] = \#\mathcal{T}$ *where* $\mathcal{T}$ *denotes the set of triangles in the graph:* $\mathcal{T} = \{\{u, v, w\} \subset [n] : uv, vw, wu \in \mathsf{edges}(G)\}$.

▷ Hint. *What is the probability that the algorithm outputs* $m(n-2)$*?*

Assume that we know a lower bound t on $\#\mathcal{T}$.

▶ ***Question 2.2)*** *Design an one-pass* (ε, δ)*-estimator for counting the number of triangles in the graph given as a stream using* $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta} \cdot \frac{mn}{t})$ *bits of memory.*

▷ Hint. *Compute the variance for the output of the previous algorithm.*

Note that it can be shown that there is no $o(n^2)$-space algorithm that approximates multiplicatively the number of triangles in a graph unless some lower bound is known on the number of triangles.