■ **Exercise 1 (A streaming algorithm for counting the number of distinct values).** [★]
We are given a *stream* of numbers $x_1, \ldots, x_n \in [m]$ and we want to compute the number of distinct values in the stream: $F_0(x) = \#\{x_i : i \in [n]\}$. (Note that if $f_a(x) = \#\{i : x_i = a\}$, we can express $F_0(x) = \sum_{a=0}^{m-1} (f_a(x))^0$, as the zero-th moment of the frequencies of each element of $[m]$ in the stream). Let us denote by $S_x = \{x_i : i \in [n]\}$ the set of the values in the stream x. Note that $F_0(x) = \#S_x$. (We may drop the x when the context is clear.)

The *streaming* constraint is that the algorithm will see every x_i only once as it reads the stream from left to right and we want to minimize the memory needed by the algorithm to accomplish this task. One can show that any deterministic algorithm that approximates the value of F_0 within 10% requires at least $\Omega(n)$ bits of memory. Here, we will design a randomized algorithm that accomplish this task using only $O(\log n + \log m)$ bits of memory.

We start with an hypothetical algorithm using uniform real random numbers and a hypothetical family of hash functions and then see how to turn it into an effective algorithm.

Assume that we are given a random function $h : [m] \to (0,1]$, i.e. such that for every $x \in [m]$, $h(x)$ is a (fixed) independent uniform random real in $(0,1]$. The algorithm proceeds as follows: when reading the stream, record in memory the minimum value μ so far of the $h(x_i)$s, and output $1/\mu - 1$ at the end.

▶ ***Question 1.1)*** *Show that* $\Pr\{\mu \geqslant t\} = (1-t)^{F_0}$.
Answer. ▷ By independence of the values of h,

$$\Pr\{\mu \geqslant t\} =_{\text{by definition of } \mu} \Pr\big\{\forall i \in [n],\ h(x_i) \geqslant t\big\} = \Pr\big\{\forall a \in S_x,\ h(a) \geqslant t\big\}$$
$$=_{\text{by independence of the } h(a)\text{s}} \prod_{a \in S_x} \Pr\{h(a) \geqslant t\} = (1-t)^{F_0}. \quad \triangleleft$$

▶ ***Question 1.2)*** *Show that* $\mathbb{E}[\mu] = \frac{1}{F_0+1}$.
Answer. ▷ As $\mu \geqslant 0$, $\mathbb{E}[\mu] = \displaystyle\int_0^\infty \Pr\{\mu \geqslant t\} dt = \int_0^1 (1-t)^{F_0} dt = \frac{1}{F_0+1}$. ◁

However, the following fact seems to imply that the algorithm is wrong.

▶ ***Question 1.3)*** *Show that* $\mathbb{E}[1/\mu] = \infty$.
Answer. ▷ Indeed, $\mathbb{E}[1/\mu] = \displaystyle\int_0^1 -\frac{d\Pr\{\mu \geqslant t\}}{t} = \int_0^1 \frac{F_0 \cdot (1-t)^{F_0-1}}{t} dt = \infty$ since $\displaystyle\frac{(1-t)^{F_0-1}}{t} \sim \frac{1}{t}$ for $t \to 0$ and $\displaystyle\int_0^\varepsilon \frac{dt}{t} = \infty$ for all $\varepsilon > 0$. ◁

But, fortunately:

▶ ***Question 1.4)*** *Compute* $\mathbb{V}\mathrm{ar}(\mu)$ *and show that* $\mathbb{V}\mathrm{ar}(\mu) \leqslant \mathbb{E}[\mu]^2$.
Answer. ▷ $\mathbb{E}[\mu^2] = \displaystyle\int_0^1 t^2 \cdot F_0 \cdot (1-t)^{F_0-1} dt = \frac{2}{(F_0+2)(F_0+1)} < 2\,\mathbb{E}[\mu]^2$. Thus, $\mathbb{V}\mathrm{ar}(\mu) = \mathbb{E}[\mu^2] - \mathbb{E}[\mu]^2 < \mathbb{E}[\mu]^2$. ◁

▶ ***Question 1.5)*** *Design and analyze a* (ε, δ)*-estimator for* F_0*. Still, what is the expected value of its output? Is there a paradox here?*
▷ Hint. *First, design an* (ε, δ)*-estimator for* μ*.*
Answer. ▷ We use the standard technics: output the median ν of $A = \lceil \alpha \ln(1/\delta) \rceil$ average of $B = \lceil \beta/\varepsilon^2 \rceil$ simultaneous independent evaluations of μ: μ_j^i for $i \in [A]$ and $j \in [B]$.

Let $\mu^i = \displaystyle\frac{\mu_1^i + \cdots \mu_B^i}{B}$. We have $\mathbb{E}[\mu^i] = \mathbb{E}[\mu] = \displaystyle\frac{1}{F_0+1}$ and $\mathbb{V}\mathrm{ar}(\mu^i) = \displaystyle\frac{\mathbb{V}\mathrm{ar}(\mu)}{B}$. Thus, by Chebyshev inequality, for all $i \in [A]$, $\Pr\left\{\left|\mu^i - \frac{1}{F_0+1}\right| \geqslant \frac{\varepsilon}{F_0+1}\right\} \leqslant \displaystyle\frac{\mathbb{V}\mathrm{ar}(\mu)/B}{\varepsilon^2/(F_0+1)^2} \leqslant \frac{1}{B \cdot \varepsilon^2} \leqslant \frac{1}{4}$ if we set $\beta = 4$.

Now, let Y_i be the indicator variable for the event $\mu^i \notin \frac{1\pm\varepsilon}{F_0+1}$. From the above, $\mathbb{E}[Y_i] \leqslant \frac{1}{4}$. But, we have $\text{Pr}\left\{\nu \notin \frac{1\pm\varepsilon}{F_0+1}\right\} \leqslant \text{Pr}\left\{\sum_{i\in[A]} Y_i \geqslant \frac{A}{2}\right\} \leqslant \text{Pr}\left\{\sum_{i\in[A]} Y_i - \sum_{i\in[A]} \mathbb{E}[Y_i] \geqslant \frac{A}{4}\right\} \leqslant_{\text{Hoeffding}} \exp\left(-\frac{2(A/4)^2}{A}\right) \leqslant \delta$ if we set $\alpha = 8$.

The (ε,δ)-estimator thus compute ν according to the above and output $1/\nu - 1$. This ensures that with probability at least $1-\delta$, the output value belongs to $[\frac{F_0}{1+\varepsilon}, \frac{F_0}{1-\varepsilon}]$ yielding a $(\varepsilon + o(\varepsilon), \delta)$-estimator for F_0.

Note that the expected value of each $1/\mu^i_j$ is still ∞ and thus the expected value of the output $1/\nu - 1$ is ∞ as well. However, with probability $1-\delta$, $1/\nu - 1$ is within ε of F_0. $\triangleleft$

Unfortunately, such a random function h requires storing m reals in memory. The key to reduce the memory needed is to relax the independence of the hash value to pairwise independence only. In the following, we will approximate the minimum of the hash keys by recording only the position of their first non-zero bit in their binary writing. We proceed as follows.

Let $\ell = \lceil \log_2 m \rceil$ such that $2^{\ell-1} < m \leqslant 2^\ell$ and consider the field with 2^ℓ elements $\mathbb{F}_{2^\ell}$. We identify $\mathbb{F}_{2^\ell}$ through canonical bijections to the set of bit-vectors $\{0,1\}^\ell$ and to the set of integers $\{0,\ldots,2^\ell-1\}$ written in binary. For every pair $(a,b) \in \mathbb{F}^2_{2^\ell}$, consider the hash function $h_{ab} : \mathbb{F}_{2^\ell} \to \mathbb{F}_{2^\ell}$ defined as $h_{ab}(y) = a + b\cdot y$. For every $y \in \mathbb{F}(2^\ell) \equiv \{0,1\}^\ell$, we denote by $\rho(y) = \max\{j \in [\ell] : y_1 = \cdots = y_j = 0\}$ the largest index j such that the first j bits of y, seen as a bit-vector, are all zero. Let us now consider the following streaming algorithm:

Algorithm 2 Streaming algorithm for F_0

```
Let $\ell = \lceil \log_2 m \rceil$, we identify each element $x_i \in [m]$ of the stream with its corresponding element in $\mathbb{F}_{2^\ell}$.
Pick uniformly and independently two random elements $a, b \in \mathbb{F}_{2^\ell}$ .
Compute $R = \max_{i=1..n} \rho(h_{ab}(x_i))$.
return $2^R$.
```

▶ ***Question* 1.6*)*** *Show that for all $c \in \mathbb{F}_{2^\ell}$ and $r \in \{0,\ldots,\ell\}$, $\Pr_{a,b}\{\rho(h_{ab}(c)) \geqslant r\} = \frac{1}{2^r}$.*

▷ Hint. *Show that $h_{ab}(c)$ is uniform in $\mathbb{F}_{2^\ell}$.*

Answer. ▷ Since a is chosen uniformly at random in $\mathbb{F}_{2^\ell}$ and independently from bc, then $a+bc$ is uniform in $\mathbb{F}_{2^\ell}$ and $h_{ab}(c)$ is an uniform random variable for all $c \in \mathbb{F}_{2^\ell}$. It follows that for all $c \in \mathbb{F}_{2^\ell}$ and $r \in \{0,\ldots,\ell\}$, the probability that the binary writing of $h_{ab}(c)$ starts with r zeros is exactly $1/2^r$. $\triangleleft$

Let W^r_c the indicator random variable for the event $\rho(h_{ab}(c)) \geqslant r$. Let$Z_r = \sum_{c\in S_x} W^r_c$, be the number of the values in the stream whose r first bits of their hash key are all zero.

▶ ***Question* 1.7*)*** *Show that $\mathbb{E}[Z_r] = F_0/2^r$.*

Answer. ▷ $\mathbb{E}[Z_r] =_{\text{linearity}} \sum_{c\in S_x} \mathbb{E}[W^r_c] =_{\text{indicator variables}} \sum_{c\in S_x} \text{Pr}\{\rho(h_{ab}(c)) \geqslant r\} = \frac{\#S_x}{2^r} = \frac{F_0}{2^r}$. $\triangleleft$

▶ ***Question* 1.8*)*** *Show that the random values $h_{ab}(0),\ldots,h_{ab}(2^\ell-1)$ are uniform and pairwise independent.*

▷ Hint. *Show that if $c \neq d$, then for all $\gamma,\delta \in \mathbb{F}_{2^\ell}$, $\Pr_{a,b}\{(h_{ab}(c), h_{ab}(d)) = (\gamma,\delta)\} = \frac{1}{\#\mathbb{F}^2_{2^\ell}}$.*

Answer. ▷ Consider $c \neq d \in \mathbb{F}_{2^\ell}$ and $(\gamma, \delta) \in \mathbb{F}_{2^\ell}^2$.

$$\Pr_{a,b}\big\{(h_{ab}(c), h_{ab}(d)) = (\gamma,\delta)\big\} = \frac{\#\{(a,b) \in \mathbb{F}_{2^\ell}^2 : (h_{ab}(c), h_{ab}(d)) = (\gamma,\delta)\}}{\#\mathbb{F}_{2^\ell}^2}$$
$$= \frac{\#\left\{(a,b) \in \mathbb{F}_{2^\ell}^2 : \begin{pmatrix} 1 & c \\ 1 & d \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}\right\}}{\#\mathbb{F}_{2^\ell}^2} = \frac{1}{\#\mathbb{F}_{2^\ell}^2},$$

since the matrix is inversible as $c \neq d$ (its determinant is $d - c$). ◁

▶ ***Question 1.9)*** *Show that* $\mathbb{V}\mathrm{ar}(Z_r) = \dfrac{F_0}{2^r}\left(1 - \dfrac{1}{2^r}\right) < \mathbb{E}[Z_r]$.

Answer. ▷ As the random variables $h_{ab}(0), \ldots, h_{ab}(2^\ell - 1)$ are pairwise independent, the random variables $(W_c^r)_{c \in S_x}$ are also pairwise independent. As the variance is linear for pairwise independent variables, we have $\mathbb{V}\mathrm{ar}(Z_r) = \sum_{c \in S_x} \mathbb{V}\mathrm{ar}(W_c^r) = \sum_{c \in S_x} \frac{1}{2^r}(1 - \frac{1}{2^r}) = \frac{F_0}{2^r}(1 - \frac{1}{2^r}) < \frac{F_0}{2^r} = \mathbb{E}[Z_r]$, since $\mathbb{V}\mathrm{ar}(\text{Bernouilli}(\alpha)) = \alpha(1-\alpha)$. ◁

Fix some $\eta > 1$.

▶ ***Question 1.10)*** *Show that* $\Pr\{Z_r > 0\} < \frac{1}{\eta}$ *for all* $r \in \{0, \ldots, \ell\}$ *such that* $2^r > \eta F_0$.
▷ Hint. *Z_r is an integer and use Markov's inequality.*
Answer. ▷ Consider r such that $2^r > \eta F_0$, i.e. such that $1/\eta > F_0/2^r = \mathbb{E}[Z_r]$. Then, $\Pr\{Z_r > 0\} = \Pr\{Z_r \geqslant 1\} \leqslant \mathbb{E}[Z_r] < 1/\eta$ by Markov's inequality. ◁

▶ ***Question 1.11)*** *Show that* $\Pr\{Z_r = 0\} < \frac{1}{\eta}$ *for all* $r \in \{0, \ldots, \ell\}$ *such that* $2^r < F_0/\eta$.
▷ Hint. *Z_r is an integer and apply Chebyshev's inequality.*
Answer. ▷ Consider r such that $2^r < F_0/\eta$, i.e. such that $\eta < F_0/2^r = \mathbb{E}[Z_r]$. Then, $\Pr\{Z_r = 0\} \leqslant \Pr\{|Z_r - \mathbb{E}[Z_r]| \leqslant \mathbb{E}[Z_r]\} \leqslant \frac{\mathbb{V}\mathrm{ar}(Z_r)}{\mathbb{E}[Z_r]^2} < 1/\mathbb{E}[Z_r] < 1/\eta$ by Chebyshev's inequality. ◁

▶ ***Question 1.12)*** *Conclude that for all* $\eta > 2$, $\Pr\{2^R \in [F_0/\eta, \eta F_0]\} > 1 - \frac{2}{\eta}$. *The algorithm outputs thus a η-approximation of F_0 with probability at least $1 - 2/\eta$ for all $\eta > 2$. How many bits of memory does it require?*
Answer. ▷ Note that $R = \max\{r : Z_r > 0\}$. Thus, for all $r \in \{0, \ldots, \ell\}$, $\Pr\{R \geqslant r\} = \Pr\{Z_r > 0\}$ and $\Pr\{R < r\} = \Pr\{Z_r = 0\}$. It follows that: with $r = \lfloor \log_2(F_0/\eta) \rfloor$, we get $\Pr\{2^R < F_0/\eta\} = \Pr\{Z_r = 0\} < 1/\eta$ by question **??**. And with $r = \lceil \log_2(\eta F_0) \rceil$, we get $\Pr\{2^R \geqslant \eta F_0\} = \Pr\{Z_r > 0\} < 1/\eta$ by question **??**. It follows that the value 2^R output by the algorithm belongs to $[F_0/\eta, \eta F_0]$ with probability at least $1 - 2/\eta > 0$, for all $\eta > 2$. The algorithm requires $2\ell + \lceil \log_2 \ell \rceil < 2\log_2 m + \log\log_2 m + 3 = O(\log m)$ bits of memory to remember a, b and R. ◁

We have thus obtained a $(\varepsilon, 2/(1+\varepsilon))$-estimator for F_0 using $O(\log m)$ bits of memory **for $\varepsilon > 1$**. Getting a (ε, δ)-estimator for F_0 in $O_{\varepsilon,\delta}(\log m + \log n)$ bits of memory for arbitrarily small $\varepsilon, \delta > 0$ requires a lot more work...