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 \( f_a(x) = \# \{ x : x = a \} \), we can express \( F_0(x) = \sum_{a=1}^{m-1} (f_a(x))^0 \), as the zero-th moment of the frequencies of each element of \([m]\) in the stream.) Let us denote by \( S_x = \{ x : x_i \in [n] \} \) the set of the values in the stream. 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 \( \mu \) so far of the \( h(x_i) \)s, and output \( 1/\mu - 1 \) at the end.

\[ \text{Exercise 1 (A streaming algorithm for counting the number of distinct values).} \]

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

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 \( \mu \) so far of the \( h(x_i) \)s, and output \( 1/\mu - 1 \) at the end.

**Question 1.1**  
Show that \( \Pr \{ \mu \geq t \} = (1 - t)^{F_0}. \)

**Answer:**  
By independence of the values of \( h, \)

\[ \Pr \{ \mu \geq t \} \text{ by definition of } \mu \Pr \{ \forall i \in [n], h(x_i) \geq t \} = \Pr \{ \forall a \in S_x, h(a) \geq t \} \]

\[ = \text{ by independence of the } h(a)\s \prod_{a \in S_x} \Pr \{ h(a) \geq t \} = (1 - t)^{F_0}. \]

**Question 1.2**  
Show that \( \mathbb{E}[\mu] = \frac{1}{F_0 + 1}. \)

**Answer:**  
As \( \mu \geq 0, \mathbb{E}[\mu] = \int_0^{\infty} \Pr \{ \mu \geq t \} dt = \int_0^{1} (1 - t)^{F_0} dt = \frac{1}{F_0 + 1}. \)

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

**Question 1.3**  
Show that \( \mathbb{E}[1/\mu] = \infty. \)

**Answer:**  
Indeed, \( \mathbb{E}[1/\mu] = \int_0^{1} \frac{d \Pr \{ \mu \geq t \} }{t} = \int_0^{1} \frac{F_0}{t} (1 - t)^{F_0 - 1} dt = \infty \) since \[ \frac{(1 - t)^{F_0 - 1}}{t} \sim \frac{1}{t} \text{ for } t \to 0 \text{ and } \int_0^{\varepsilon} \frac{dt}{t} = \infty \text{ for all } \varepsilon > 0. \]

But, fortunately:
**Question 1.4** Compute $\text{Var}(\mu)$ and show that $\text{Var}(\mu) \leq E[\mu]^2$.

**Answer.**

$$E[\mu^2] = \int_0^1 t^2 \cdot F_0 \cdot (1 - t)^{F_0 - 1} dt = \frac{2}{(F_0 + 2)(F_0 + 1)} < 2E[\mu]^2.$$ Thus, $\text{Var}(\mu) = E[\mu^2] - E[\mu]^2 < E[\mu]^2$.

**Question 1.5** Design and analyze a $(\varepsilon, \delta)$-estimator for $F_0$. Still, what is the expected value of its output? Is there a paradox here?

**Hint.** First, design an $(\varepsilon, \delta)$-estimator for $\mu$.

**Answer.**

We use the standard techniques: output the median $\nu$ of $A = \{ \alpha \ln(1/\delta) \}$ average of $B = \{ \beta / \varepsilon^2 \}$ simultaneous independent evaluations of $\mu$: $\mu_i^j$ for $i \in [A]$ and $j \in [B]$.

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

Now, let $Y_i$ be the indicator variable for the event $\mu^i \not\in \left[ \frac{1+\varepsilon}{F_0 + 1} \right]$. From the above, $E[Y_i] \leq \frac{1}{4}$. But, we have $Pr \left( \nu \not\in \left[ \frac{1}{F_0 + 1} \right] \right) \leq Pr \left( \sum_{i \in [A]} Y_i \geq \frac{A}{4} \right) \leq \text{Hoeffding exp} \left( -\frac{2(A/4)^2}{A} \right) \leq \delta$ if we set $\alpha = 8$.

The $(\varepsilon, \delta)$-estimator thus compute $\nu$ 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 $\infty$ and thus the expected value of the output $1/\nu - 1$ is $\infty$ as well. However, with probability $1 - \delta$, $1/\nu - 1$ is within $\varepsilon$ of $F_0$.

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 $2^{\ell - 1} < m \leq 2^{\ell}$ and consider the field with $2^{\ell}$ elements $F_{2^\ell}$. We identify $F_{2^\ell}$ through canonical bijections to the set of bit-vectors $\{0, 1\}^\ell$ to the set of integers $\{0, \ldots, 2^\ell - 1\}$ written in binary. For every pair $(a, b) \in F_{2^\ell}^2$, consider the hash function $h_{ab} : F_{2^\ell} \to F_{2^\ell}$ defined as $h_{ab}(y) = a + b \cdot y$. For every $y \in F(2^\ell) = \{0, 1\}^\ell$, we denote by $p(y) = \max \{ j \in [\ell] : y_1 = \cdots = y_j = 0 \}$ 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 $x \in [m]$ of the stream with its corresponding element in $F_{2^\ell}$.

Pick uniformly and independently two random elements $a, b \in F_{2^\ell}$.

Compute $R = \max_{i=1..n} p(h_{ab}(x_i))$.

return $2^R$.

**Question 1.6** Show that for all $c \in F_{2^\ell}$ and $r \in \{0, \ldots, \ell\}$, $Pr_{a,b} \left( \rho(h_{ab}(c)) \geq r \right) = \frac{1}{2^r}$.

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

Show that $1.10$.

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

**Answer:**

$$\frac{\sum_{c \in S_x} \mathbb{E}[W_c^r]}{2^r} = \frac{F_0}{2^r} \quad \triangleleft$$

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

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

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

$$\mathbb{Pr}_{a,b}\{ (h_{ab}(c), h_{ab}(d)) = (\gamma, \delta) \} = \frac{\# \{(a, b) \in \mathbb{F}_{2^k}^2 : (h_{ab}(c), h_{ab}(d)) = (\gamma, \delta) \}}{\# \mathbb{F}_{2^k}^2} = \frac{1}{\# \mathbb{F}_{2^k}^2},$$

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

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

**Answer:**

As the random variables $h_{ab}(0), \ldots, h_{ab}(2^k - 1)$ 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 $\mathbb{V}[Z_r] = \sum_{c \in S_x} \mathbb{V}[W_c^r] = \sum_{c \in S_x} \frac{1}{2^r} (1 - \frac{1}{2^r}) = \frac{F_0}{2^r} (1 - \frac{1}{2^r}) < \frac{F_0}{2^r} = \mathbb{E}[Z_r]$, since $\mathbb{V}[	ext{Bernouilli}(\alpha)] = \alpha(1-\alpha)$. \triangleleft

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, $\Pr\{ Z_r > 0 \} = \Pr\{ Z_r \geq 1 \} \leq \mathbb{E}[Z_r] < 1/\eta$ by Markov’s inequality. \triangleleft

**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, $\Pr\{ Z_r = 0 \} \leq \Pr\{ |Z_r - \mathbb{E}[Z_r]| \geq \mathbb{E}[Z_r] \} \leq \frac{\mathbb{V}[Z_r]}{\mathbb{E}[Z_r]^2} < 1/\mathbb{E}[Z_r] < 1/\eta$ by Chebyshev’s inequality. \triangleleft

**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 $\eta$-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 $r \in \{0, \ldots, \ell\}$, $\Pr\{ R \geq r \} = \Pr\{ Z_r > 0 \}$ and $\Pr\{ R < r \} = \Pr\{ Z_r = 0 \}$. It follows that: with $r = \lceil \log_2(F_0/\eta) \rceil$, we get $\Pr\{ 2^r < F_0/\eta \} = \Pr\{ Z_r = 0 \} < 1/\eta$ by question 1.1.
And with \( r = [\log_2(\eta F_0)] \), we get \( \Pr\{2^R \geq \eta F_0\} = \Pr\{Z_x > 0\} < 1/\eta \) by question 1.4. 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 \( 2\ell + [\log_2 \ell] \leq 2 \log_2 m + \log \log_2 m + 3 = O(\log m) \) bits of memory to remember \( a, b \) and \( R \).<\n
We have thus obtained a \((\varepsilon, 2/(1 + \varepsilon))\)-estimator for \( F_0 \) using \( O(\log n) \) bits of memory for \( \varepsilon > 1 \). Getting a \((\varepsilon, \delta)\)-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...

\textbf{Exercise 2 (Generating function for the Galton-Watson population total number).} Let \( Z \) be the random variable for the total number of nodes in a Galton-Watson branching process for which the extinction probability is 1: \( Z = \sum_{i,n} Z_i^{(n)} \), where \( Z_i^{(n)} \sim Z \) denotes the number of children of the \( i \)th node on level \( n \). Let \( g_Z \) be its generating function.

\textbf{Question 2.1} Show that \( g_Z(s) = sg_Z(g_Z(s)) \).

\textbf{Answer} \( g_Z(s) = \mathbb{E}[s^Z] \)

\begin{align*}
&= \sum_{k=0}^{\infty} \Pr\{Z = k\} \mathbb{E}[s^Z \mid Z_1^{(0)} = k] \\
&= \sum_{k=0}^{\infty} \Pr\{Z = k\} \sum_{j=0}^{\infty} s^{1+\sum_{i=1}^{j} Z_i^{(0)}} \Pr\{Z_1^{(0)} = k\} \\
&= \sum_{k=0}^{\infty} \Pr\{Z = k\} s \mathbb{E}[s^Z]^k \\
&= sg_Z(g_Z(s)).
\end{align*}

\textbf{Exercise 3 (Branching processes in continuous time \( \star \))}. Recall that an exponential random variable \( X \) with parameter \( \lambda > 0 \) is defined by: \((\forall x \geq 0) \Pr\{X \geq x\} = e^{-\lambda x}\).

Consider the following process:

- At time 0, \( Z_0 = 1 \) (the root of the process). By convention, this node is born at time 0.
- When a node \( i \) is born, its lifetime is exponentially distributed with parameter \( \mu \): if it is born at time \( t \), it dies at time \( t + U_i \), where \( U_i \) is exponentially distributed with parameter \( \mu \).
- An alive node \( i \) can give birth to children. Its children birthdates are generated according to an exponential distribution with parameter \( \lambda \): if a node is born at time \( t \), its first child is generated at time \( t + V_i^1 \) (if it is not dead before), its second child at time \( t + V_i^1 + V_i^2 \), and so on where \( V_i^j \) is exponentially distributed with parameter \( \lambda \).
- All the lifetimes \( (U_i) \) and birth intervals \( (V_i^j) \) form a mutually independent family of random variables.

\textbf{Question 3.1} Show that the exponential distribution is memoryless, i.e. if \( X \) is exponentially distributed with parameter \( \lambda \), then \((\forall t, u \geq 0) \Pr\{X \geq t + u \mid X \geq t\} = \Pr\{X \geq u\}\).

\textbf{Answer} \( \Pr\{X \geq t + u \mid X \geq t\} = \frac{\Pr\{X \geq t + u \land X \geq t\}}{\Pr\{X \geq t\}} = \frac{\Pr\{X \geq t + u\}}{\Pr\{X \geq t\}} = e^{-\lambda(t+u)+\lambda t} = e^{-\lambda u} = \Pr\{X \geq u\} \).<\n
4
Let $X$ and $Y$ be two independent exponentially distributed random variables with respective parameters $\lambda$ and $\mu$.

**Question 3.2** Show that $\min(X, Y)$ is also exponentially distributed. What is its parameter?

**Answer.** Let $M = \min(X, Y)$, then:

$$
\Pr\{M \geq t\} = \Pr\{X \geq t \wedge Y \geq t\} = \Pr\{X \geq t\} \cdot \Pr\{Y \geq t\}, \quad \text{by independence of } X \text{ and } Y
$$

$$
= e^{-(\lambda+\mu)t}.
$$

$M$ is thus exponentially distributed with parameter $\lambda + \mu$. <

**Question 3.3** What is the probability that $\min(X, Y) = X$?

**Answer.** Let $M = \min(X, Y)$, then:

$$
\Pr\{M = X\} = \int_{t=0}^{\infty} \Pr\{Y > t \wedge X \in [t, t+dt]\}
$$

$$
= \int_{t=0}^{\infty} \Pr\{Y > t\} \cdot \Pr\{X \in [t, t+dt]\}, \quad \text{by independence of } X \text{ and } Y
$$

$$
= \int_{t=0}^{\infty} e^{-\mu t} \cdot \lambda e^{-\lambda t} dt = \frac{\lambda}{\lambda + \mu}.
$$

<

We are back to the branching process.

**Question 3.4** What is the law of the number of children for each node?

**Answer.** Consider w.l.o.g. a node born at time $0$. Assume it has already $k$ children and that its last child is born at time $t$. Since its death date $U \geq t$ is exponentially distributed, it is memory less and it is distributed as $t + U \sim t + \text{Exp}(\mu)$. Its potential next child birth is scheduled at time $t + V$ where $V \sim \text{Exp}(\lambda)$. It follows that the probability that it has a $(k+1)$th child before dying is $\Pr\{\min(U, V) = V\} = \lambda/(\lambda + \mu)$, independent of $k$. Now, let $p_k$ be the probability that a given node has $k$ children. We have $p_0 = \Pr\{\min(U, V) = U\} = \mu/(\lambda + \mu)$. And as, in order to get $(k+1)$ children, it must first have one child and then $k$ children and die, and because the process is memoryless, we have $p_{k+1} = \lambda/(\lambda + \mu) \cdot p_k = \mu \lambda^{k} / (\lambda + \mu)^{k+1}$. Thus, the number of children of a given node is a random variable $Z$ distributed exponentially and independently as: $\Pr\{Z = k\} = \mu \lambda^{k} / (\lambda + \mu)^{k+1}$. <

**Question 3.5** What is the probability of extinction of this process?

**Answer.** The generating function for $Z$ is:

$$
g_Z(x) = \frac{\mu}{\lambda + \mu} \sum_{k=0}^{\infty} \left(\frac{\lambda x}{\lambda + \mu}\right)^k = \frac{\mu}{\lambda + \mu} \cdot \frac{1}{1 - \lambda x / (\lambda + \mu)} = \frac{\mu}{\lambda + \mu - \lambda x}
$$

It follows that $E[Z] = g'_Z(1) = \lambda \mu / (\lambda + \mu - \lambda)^2 = \lambda / \mu$. Note that $\Pr\{Z = 0\} > 0$ for all $\lambda, \mu > 0$. Thus, according to the lecture, the process gets extinct almost surely when $\lambda \leq \mu$. When $\lambda > \mu$, then the extinction probability is the non-one solution to $p = g_Z(p)$, i.e. $p(\lambda + \mu - \lambda p) = \mu$. As 1 is solution of this equation, we can factor it as: $(p - 1)(\mu - \lambda p) = 0$. The extinction probability is thus $\min(1, \mu / \lambda)$. <