■ **Exercise 1 (Missing element & distinct elements).** Assume we are reading a stream of n distinct integers in $\{1,\ldots,n+1\}$.

▶ ***Question* 1.1*)*** *Assume first that all of the elements in the stream are indeed distinct elements of $\{1,\ldots,n+1\}$) and design for this case a deterministic $O(\log n)$ bits-memory algorithm that outputs the missing element.*

Let us now waive the assumption that the integers are distinct and let us design an algorithm to check this property.

▶ ***Question* 1.2*)*** *Consider a prime number $p \geqslant n^2$ and a non-zero polynomial $U(X)$ of degree at most n over the field $\mathbb{Z}_p$. Show that $\Pr_a\{U(a) = 0 \bmod p\} \leqslant \frac{1}{n}$ when a is chosen uniformly at random in $\mathbb{Z}_p$.*
▷ Hint. *How many solutions are there to $U(a) = 0$ in the field $\mathbb{Z}_p$?*

Consider the following algorithm: Pick a prime number p such that $n^2 \leqslant p < 2n^2$ (there is always one). Pick an integer $a \in \{0,\ldots,p-1\}$ uniformly at random. Compute $S := \sum_{i=1}^{n} x_i$, $y := \frac{(n+1)(n+2)}{2} - S$, $U := \sum_{i=1}^{n} a^{x_i-1} \bmod p$ and $V := \sum_{i=0}^{n} a^i \bmod p$. If $U == V - a^{y-1} \bmod p$, then answer « *y is the missing element* », and answer « *the stream does not contain n distinct integers in $\{1,\ldots,n+1\}$* » otherwise.

▶ ***Question* 1.3*)*** *Show that this is a $O(\log n)$ bits-memory streaming algorithm that always outputs the right answer when the stream matches the specification, and that detects every erroneous stream with probability at least $1-1/n$.*

■ **Exercise 2 (Traffic monitoring: uniformity detection).** Imagine that we are running a huge website and we want to prevent attacks by keeping track of the origins of the various clients currently connected to the server. Along time, clients connect and then disconnect from the website. And we want to detect if all the clients connected are from the same IP address. But we do not want to slow down the server and wish to dedicate to this task only a *constant* memory, i.e. only a constant number of integers. We model the problem as follows:

We are given an infinite stream of events $e_1, e_2, \ldots, e_n, \ldots$ where each e_i is either **connect**(x) or **disconnect**(x) where x is a positive integer standing for the IP address of the client (dis-)connecting. We assume that the stream is wellformed, i.e. that there are always at least as many events **connect**(x) as **disconnect**(x) from the beginning of the stream to any position for every integer x. We want to detect when all the clients connected have the same IP address x.

▶ ***Question* 2.1*)*** *Spot when to set the alarm on in the following sequence where x denotes the event* **connect**(x) *and $\bar{x}$ the event* **disconnect**(x)*:*

$$1, 2, 3, \bar{2}, \bar{3}, 1, 1, \bar{1}, 4, 6, 7, \bar{1}, \bar{6}, \bar{1}, 2, \bar{2}, \bar{4}, 8, 3, \bar{3}, \bar{7}, 9$$

We consider the following algorithm that uses only three integer variables:

- start with $\mathtt{n} := 0$, $\mathtt{a} := 0$ and $\mathtt{b} := 0$ at $t = 0$;
- on event **connect**(x): do $\mathtt{n} := \mathtt{n}+1$, $\mathtt{a} := \mathtt{a}+x$ and $\mathtt{b} := \mathtt{b}+x^2$;
- on event **disconnect**(x): do $\mathtt{n} := \mathtt{n}-1$, $\mathtt{a} := \mathtt{a}-x$ and $\mathtt{b} := \mathtt{b}-x^2$;
- set on the alarm every time that $\mathtt{n} > 0$ and $\mathtt{b} = \mathtt{a}^2/\mathtt{n}$.

The right way to the correctness of this deterministic algorithm passes through the analysis of a random variable. Consider a random variable X taking positive integer values. We denote by $\mathrm{supp}(X) = \{\, x : \Pr\{X = x\} > 0 \,\}$ and assume that $|\mathrm{supp}(X)| < \infty$. We denote by $\mathbb{E}[X]$ and $\mathbb{V}\mathrm{ar}[X] = \mathbb{E}[(X - \mathbb{E}(X))^2]$ respectively the expectation and the variance of X.

▶ ***Question* 2.2*)*** *Show that $|supp(X)| = 1$ if and only if $\mathbb{V}\mathrm{ar}[X] = 0$.*

▶ ***Question* 2.3*)*** *Conclude that the algorithm is correct.*

■ **Exercise 3 (((ε, δ)-estimator).** Suppose we want to compute a value μ from some data. Assume that we have a randomized algorithm A that computes a random variable Z such that $\mathbb{E}[Z] = \mu$ and $\mathbb{V}\text{ar}(Z) \leqslant A \cdot \mu^2$ for some constant $A > 0$.

▶ ***Question* 3.1*)*** *Design a (ε, δ)-estimator for μ for all $\varepsilon > 0$ and $\delta > 0$ making $O(\frac{\log(1/\delta)}{\varepsilon^2})$ calls to the randomized algorithm A. Give exact bounds on the number of calls, explain how you proceed.*

Assume now that we have an algorithm B that computes a random variable Y such that $\Pr\{Y \notin [\frac{\mu}{a}, a \cdot \mu]\} \leqslant b$ for some $a > 1$ and $b < \frac{1}{2}$. (No assumption is made on $\mathbb{E}[Y]$, it might even be $\neq \mu$)

▶ ***Question* 3.2*)*** *Can we design a (ε, δ)-estimator for μ using algorithm B for all $\varepsilon > 0$ and $\delta > 0$? If not, what are the values of ε and δ for which we can design a (ε, δ)-estimator and how do you proceed?*

■ **Exercise 4 (Pairwise independent random bits).** We will describe a way to generate n pairwise independent uniform random bits $X_1, \ldots, X_n$ using only $\ell = \lceil \log_2 n \rceil$ "true" uniform independent random bits $Y_1, \ldots, Y_\ell$.

▶ ***Question* 4.1*)*** *Let $(G, \cdot)$ be a finite group, X a random variable over G and U an independent uniform random variable over G. Show that $X \cdot U$ is an uniform random variable over G independent from X.*

Let $[i] = \{j : j\text{-th bit of } i \text{ written in binary is } 1\} \subseteq \{1, \ldots, \ell\}$ such that $i = \sum_{j \in [i]} 2^{j-1}$ for all $i \in \{1, \ldots, n\}$. Consider $Y_1, \ldots, Y_\ell$, ℓ uniform independent random bits. We then set $X_i = \bigoplus_{j \in [i]} Y_j$ for $i = 1 \ldots n$, where $a \oplus b$ denote the XOR of a and b (i.e. their sum modulo 2). For instance: $13 = 1101$ in binary, thus $X_{13} = Y_4 \oplus Y_3 \oplus Y_1$.

▶ ***Question* 4.2*)*** *Show that $X_1, \ldots, X_n$ are n pairwise independent uniform random bits.*