Randomized Methods in Computation*
Lecture 12: Hitters and Samplers

Notes taken by Amos Korman, Yoav Rodeh, Eran Tromer

Spring 2001

Summary: A *sampler* is an oracle Turing machine that estimates the average of any function $f : \{0,1\}^n \to [0,1]$ with bounded deviation and bounded probability of failure. A *hitter* is an oracle Turing machine that, given a boolean function $f : \{0,1\}^n \to \{0,1\}$, finds x s.t. $f(x) = 1$ with a bounded probability of failure if f has value 1 on at least a constant fraction of $\{0,1\}^n$. These are fundamental and useful problems. We consider the randomness and query complexities of hitters and samplers whose running time is polynomially bounded, and recall lower bounds for each. We show several simple constructions for both, as well as improved constructions based on pairwise-independent sample spaces and on random walks on expander graphs. We then show composite constructions whose complexities match the lower bounds.

1 Definitions and Lower Bounds

1.1 Definitions

Suppose we are given black-box access to a function $f : \{0,1\}^n \to [0,1]$ or $f : \{0,1\}^n \to \{0,1\}$. A problem of interest is to estimate its average value:

$$\bar{f} \stackrel{\text{def}}{=} \sum_{x\in\{0,1\}^n} f(x)$$

An exact solution would require evaluating f at all points, which takes time exponential in n. However, an approximate solution could be quite useful: for instance, given a $\mathcal{BPP}$ algorithm that has error probability bounded by $\frac{1}{3}$, estimating $\bar{f}$ within error smaller than $\frac{1}{6}$ would suffice to obtain a decision with certainty.

By relaxing the requirement further to allow some probability of failure, we obtain the notion of a sampler. Given oracle access to $f : \{0,1\}^n \to [0,1]$ the sampler outputs a value that is within ε of $\bar{f}$ with probability at least $1-\delta$. A boolean sampler is similar but the domain of f is limited to $\{0,1\}$. Obviously any sampler is a boolean sampler, but the converse is not true.

Throughout the discussion we assume that the values of f are given with at least $\log_2(1/\varepsilon)$ bits of precision (otherwise samplers don't exist), but no more than $\text{poly}(\log(1/\varepsilon), \log(1/\delta), n)$ bits (otherwise merely reading the oracle tape would exceed the polynomial time bound discussed below).

* Lecture Notes for a course given by Oded Goldreich at the Weizmann Institute of Science, Israel.

Definition 1 (Sampler) *A probabilistic oracle Turing machine M is a* sampler *if for any $n \in \mathsf{N}$, $\varepsilon \in (0,1]$ and $\delta \in (0,1]$ and for every function $f : \{0,1\}^n \to [0,1]$:*

$$\Pr\left[\left|M^f(\varepsilon,\delta,n) - \bar{f}\right| > \varepsilon\right] < \delta$$

Definition 2 (Boolean Sampler) *A probabilistic oracle Turing machine M is a* boolean sampler *if for any $n \in \mathsf{N}$, $\varepsilon \in (0,1]$ and $\delta \in (0,1]$ and for every function $f : \{0,1\}^n \to \{0,1\}$:*

$$\Pr\left[\left|M^f(\varepsilon,\delta,n) - \bar{f}\right| > \varepsilon\right] < \delta$$

Another scenario is this: we are given black-box access to $f : \{0,1\}^n \to \{0,1\}$ and are assured that it has value 1 on at least an ε fraction of its domain. The problem is to find one of these values, i.e., to find a representative of the "good set" $f^{-1}(1)$. As before, in the worst case we could check an exponential number of values without succeeding. By allowing some probability of error we obtain the notion of a hitter.

For $f : \{0,1\}^n \to \{0,1\}$ and $a \in \{0,1\}$, denote:

$$f^{-1}(a) \stackrel{\text{def}}{=} \{x \in \{0,1\}^n \mid f(x) = a\}$$

For $\varepsilon \in [0,1]$, denote:

$$F(n,\varepsilon) \stackrel{\text{def}}{=} \left\{f \;\mid\; f : \{0,1\}^n \to \{0,1\},\ |f^{-1}(1)| \geq \varepsilon \cdot 2^n\right\}$$

That is, $F(n,\varepsilon)$ the set of functions that have value 1 on at least an ε fraction of their domain.

Definition 3 (Hitter) *A probabilistic oracle Turing machine M is a* hitter *if for any $n \in \mathsf{N}$, $\varepsilon \in (0,1]$ and $\delta \in (0,1]$ and for any function $f \in F(n,\varepsilon)$:*

$$\Pr\left[M^f(\varepsilon,\delta,n) \in f^{-1}(1)\right] > 1 - \delta$$

For both samplers and hitter, the input δ specifies the allowed error probability. The input ε differs: for samplers it specifies the precision required of the answer, and for hitters it specifies a promise made on the input function.

In Subsection 2.1, we show that any sampler yields a boolean hitter with related complexity.

1.2 Complexity Measures

As suggested above, the motivation for the relaxed definition is the ability to reduce complexity. For any probabilistic oracle Turing machine M (specifically, for samplers, boolean samplers and hitters), we are interested in the following:

Time complexity $T_M(\varepsilon,\delta,n)$: We require $T_M(\varepsilon,\delta,n) = \mathrm{poly}\left(n, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$. That is, M should halt within time polynomial in n, $\frac{1}{\varepsilon}$ and $\log\left(\frac{1}{\delta}\right)$.

Query complexity $Q_M(\varepsilon,\delta,n)$: The maximum number of queries made by $M^f(\varepsilon,\delta,n)$, taken over all possible coin tosses and all possible oracles $f \in F(n,\varepsilon)$. We wish to minimize this, because evaluation may be expensive or the context may bound the number of permissible evaluations.

Randomness complexity $R_M(\varepsilon, \delta, n)$**:** The maximum number of probabilistic bits used by $M^f(\varepsilon, \delta, n)$, i.e., the number of binary coin flips it may use, taken over all possible oracles $f \in F(n, \varepsilon)$. We wish to minimize this parameter as well: for small δ we may need to try many different random choices, so it is useful to reduce the space of this choices (we'll see an example of this in Section 4).

1.3 Lower Bounds

The following lower bounds on complexity hold:

	$Q_M(\varepsilon, \delta, n) \geq$	$R_M(\varepsilon, \delta, n) \geq$
Hitters	$\Omega\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$	$n + \log_2\left(\frac{1}{\delta}\right) - \log_2(Q_M(\varepsilon, \delta, n)) - O(1)$
Samplers	$\Omega\left(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right)$	$n + \log_2\left(\frac{1}{\delta}\right) - \log_2(Q_M(\varepsilon, \delta, n)) - O(1)$

We proceed to prove the lower bounds for hitters[1]. For proofs of the lower bounds on the complexity of samplers, see [2] or the weaker results of Subsection 2.1.

The lower bounds which we derive on the query and randomness complexities of hitters apply only when ε, as a function of n and δ, is upper bounded by some constant $c < 1$. Some limitation must indeed be imposed on ε, since when ε approaches 1 very quickly it's possible that few queries or little randomness are required. For instance, consider the case $\varepsilon = 1 - \frac{1}{2^n}$, i.e., $|f^{-1}(0)| = 1$. In this case, for any fixed δ and sufficiently large n, a *single* random query will hit $f^{-1}(1)$ with probability greater than δ. Similarly, when $|f^{-1}(0)| = 1$ evaluating any two *fixed* elements guarantees success regardless of δ, using no randomness.

For the bounds to hold we also need to restrict the values of δ, as the following shows: for given n and ε, the minimum query complexities is no more than $(1 - \varepsilon) \cdot 2^n$, regardless of δ. Similarly, an algorithm that is allowed to make 2^n queries needs no randomness, regardless of δ.

We will need the following technical lemma:

Lemma 1.1 *For any constant $c \in (0, 1)$, there exists a constant $c' > 0$ such that for any $N, s \in \mathsf{N}$ and $\varepsilon \in (0, c)$ such that $(1 - \varepsilon) \cdot N > s$:*

$$\frac{\binom{N-s}{\varepsilon \cdot N}}{\binom{N}{\varepsilon \cdot N}} > e^{-c' \cdot s \cdot \varepsilon}$$

Moreoever, when $\varepsilon \ll 1$, $s \ll N$, $\varepsilon \cdot N \gg 1$ and $(1 - \varepsilon) \cdot N - s \gg 1$: $c' \approx 2$.

Proof: According to Stirling's formula: $k! \approx \sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ for $k \gg 1$. This yields the following approximation for the binomial coefficient $\binom{a}{b}$ when b and $(a - b)$ are large:

$$\binom{a}{b} = \frac{a!}{b!(a-b)!} \approx \frac{\sqrt{2\pi a}\left(\frac{a}{e}\right)^a}{\sqrt{2\pi b}\left(\frac{b}{e}\right)^b \cdot \sqrt{2\pi(a-b)}\left(\frac{a-b}{e}\right)^{a-b}} = \sqrt{\frac{a}{2\pi \cdot b \cdot (a-b)}} \cdot \left(\frac{a}{b}\right)^b \cdot \left(\frac{a}{a-b}\right)^{a-b}$$

Denoting $\sigma = \frac{s}{N}$ and cancelling out N factors, we get:

$$\frac{\binom{N-s}{\varepsilon \cdot N}}{\binom{N}{\varepsilon \cdot N}} \approx \frac{\sqrt{\frac{1-\sigma}{N \cdot 2\pi \cdot \varepsilon \cdot (1-\sigma-\varepsilon)}} \cdot \left(\frac{1-\sigma}{\varepsilon}\right)^{\varepsilon \cdot N}\left(\frac{1-\sigma}{1-\sigma-\varepsilon}\right)^{N-s-\varepsilon \cdot N}}{\sqrt{\frac{1}{N \cdot 2\pi \cdot \varepsilon \cdot (1-\varepsilon)}} \cdot \left(\frac{1}{\varepsilon}\right)^{\varepsilon \cdot N}\left(\frac{1}{1-\varepsilon}\right)^{N-\varepsilon \cdot N}} = \frac{\sqrt{\frac{1-\sigma}{(1-\sigma-\varepsilon)}}}{\sqrt{\frac{1}{(1-\varepsilon)}}} \cdot \frac{\left(\frac{1-\sigma}{\varepsilon}\right)^{\varepsilon \cdot N}}{\left(\frac{1}{\varepsilon}\right)^{\varepsilon \cdot N}} \cdot \frac{\left(\frac{1-\sigma}{1-\sigma-\varepsilon}\right)^{N-s-\varepsilon \cdot N}}{\left(\frac{1}{1-\varepsilon}\right)^{N-\varepsilon \cdot N}}$$

[1]The proofs in this section are provided by the scribes and did not appear in the lecture.

Note that $\frac{1-\sigma}{1-\sigma-\varepsilon} > \frac{1}{1-\varepsilon}$, hence for some constants $c_1, c_2 < 1$ and c', we get the lower bound:

$$\frac{\left(\frac{1-\sigma}{\varepsilon}\right)^{\varepsilon \cdot N}}{\left(\frac{1}{\varepsilon}\right)^{\varepsilon \cdot N}} \cdot \frac{\left(\frac{1}{1-\varepsilon}\right)^{N-s-\varepsilon \cdot N}}{\left(\frac{1}{1-\varepsilon}\right)^{N-\varepsilon \cdot N}} = (1-\sigma)^{\varepsilon \cdot N} \cdot (1-\varepsilon)^{s} = (1-\sigma)^{\frac{1}{\sigma} \cdot s \cdot \varepsilon} \cdot (1-\varepsilon)^{\frac{1}{\varepsilon} \cdot s \cdot \varepsilon} > (c_1)^{s \cdot \varepsilon} \cdot (c_2)^{s \cdot \varepsilon} = e^{-c' \cdot s \cdot \varepsilon}$$

where the last inequality relies on ε and $\sigma = \frac{s}{N}$ being upper bounded by constants smaller than 1. The approximations (Stirling's formula) can be accounted for by a small increase of c' (rigorous analysis is possible using strict bounds instead of approximations). The main part of the lemma follows.

Moreover, the additional conditions listed in the lemma imply that the approximations are very accurate, and furthermore $\varepsilon, \sigma \ll 1$ implies $c_1, c_2 \approx \frac{1}{e}$, hence $c' \approx 2$.

■

Theorem 4 *For any constant $c \in (0,1)$: for any polynomial time hitter H and any $n \in \mathsf{N}$, $\varepsilon \in (0,c)$ and $\delta \in (0,\ e^{-\varepsilon \cdot (1-\varepsilon) \cdot 2^n})$,*

$$Q_H(\varepsilon, \delta, n) \geq \Omega\left(\frac{1}{\varepsilon} \log\left(\frac{1}{\delta}\right)\right)$$

Proof: Using Yao's Lemma [3], to prove the lower bound on probabilistic algorithms it suffices to show a probability distribution Φ on $F(n,\varepsilon)$ such that every deterministic oracle Turing machine D needs $\Omega\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$ queries to hit a good element with probability at least δ, where the probability is taken over the distribution Φ.

We define Φ to be the uniform distribution over all function $f \in F(n)$ s.t. $|f^{-1}(1)| = \lceil \varepsilon \cdot 2^n \rceil$.

Let D be a a deterministic oracle Turing machine D that succeeds with probability at least δ. Define $\mathit{Samples}_D(\varepsilon, \delta, n)$ to be the set of all queries made by $D^f(\varepsilon, \delta, n)$ when D receives only "0" answers to its oracle queries. This set depends only on the inputs n, ε and δ, and not on the function f. Let $s = |\mathit{Samples}_D(\varepsilon, \delta, n)|$.

If $s \geq \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$ then we're done. Otherwise: the limited domain of δ implies $\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right) < (1-\varepsilon) \cdot 2^n$, so we have $s < (1-\varepsilon) \cdot 2^n$. The probability that D does not succeed is

$$\begin{aligned} & \Pr\left[f^{-1}(1) \cap \mathit{Samples}_D(\varepsilon, \delta, n) = \emptyset\right] \\ = & \ \frac{\#\text{ ways to choose } \varepsilon \cdot 2^n \text{ elements outside } \mathit{Samples}_D(\varepsilon, \delta, n)}{\#\text{ ways to choose } \varepsilon \cdot 2^n \text{ elements}} \\ = & \ \frac{\binom{2^n - s}{\varepsilon \cdot 2^n}}{\binom{2^n}{\varepsilon \cdot 2^n}} > e^{-c' \cdot \varepsilon \cdot s} \end{aligned}$$

for some constant $c' < 1$, where the inequality follows from Lemma 1.1 using $N = 2^n$.

The probability $e^{-c' \cdot \varepsilon \cdot s}$ must be smaller than δ, and $s \leq Q_D(\varepsilon, d, n)$ by the definition of query complexity, so we obtain:

$$Q_D(\varepsilon, d, n) \geq s \geq \Omega\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$$

Note that when the conditions of Lemma 1.1 hold (and in particular ε is small), the constant hidden in the Ω notation is approximately 2. ■

Theorem 5 *For any polynomial time hitter H and constant $\varepsilon \in (0,1)$ s.t. $Q_H(\varepsilon,\delta,n) < (1-\varepsilon)\cdot 2^n$:*

$$R_H(\varepsilon,\delta,n) \geq n + \log_2\left(\frac{1}{\delta}\right) - \log_2(Q_H(\varepsilon,\delta,n)) - O(1)$$

Proof: Fix ε, δ and n. Denote by r the maximum number of random bits used by $H^f(\varepsilon,\delta,n)$, taken over all possible $f \in F(n,\varepsilon)$.

First we show that $\delta \geq \frac{1}{2^r}$. If this does not hold then H answers correctly for any random seed, so by fixing its random choices arbitrarily we obtain a deterministic hitter H' with query complexity $Q_{H'}(\varepsilon,\delta,n) < (1-\varepsilon)\cdot 2^n$. This is impossible since H' fails when given the following oracle $f_{H'}(x) \in F(n,\varepsilon)$:

$$f_{H'}(x) = \begin{cases} 0 & \text{if } H' \text{ queries } x \\ 1 & \text{otherwise} \end{cases}$$

We now turn to the heart of the analysis (of H). For $w \in \{0,1\}^r$, let $\mathit{Samples}(w) \subseteq \{0,1\}^n$ denote the set of all queries made by H with w as its random coin tosses when H receives only "0" anwers to its oracle queries. For $U \subseteq \{0,1\}^r$, define $\mathit{Samples}(U) = \bigcup_{w\in U} \mathit{Samples}(w)$, and define the function $f_U : \{0,1\}^n \to \{0,1\}$ such that

$$f_U(x) = \begin{cases} 0 & \text{if } x \in \mathit{Samples}(U) \\ 1 & \text{otherwise} \end{cases}$$

If the random coin toss of H is $w \in U$ then given f_U as an oracle, H receives only "0" answers and thus fails. Since the coin toss of H is in U with probability at least $\frac{1}{2^r}|U|$, this yields a lower bound on the probability of failure [2]:

$$\Pr\left[H^{f_U} \notin f_U^{-1}(1)\right] > \frac{1}{2^r}|U| \tag{1}$$

Denote by q the maximum number of samples made by H^f taken over all $f \in F(n,\varepsilon)$ and random coin tosses. Then, for every w, $|\mathit{Samples}(w)| \leq q$. Therefore, $\mathit{Samples}(U) \leq q\cdot|U|$, and so:

$$|f_U^{-1}(1)| = 2^n - |\mathit{Samples}(U)| \geq 2^n - q\cdot|U|$$

Setting $|U| = \left\lfloor \frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n \right\rfloor$, we get

$$|f_U^{-1}(1)| \geq 2^n - q\cdot|U| = 2^n - q\cdot\lfloor\frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n\rfloor \geq 2^n - q\cdot\frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n = \varepsilon\cdot 2^n$$

Thus, $f_U \in F(n,\varepsilon)$. H is a hitter, so according to (1) we have $\frac{1}{2^r}|U| < \delta$, i.e.,

$$\frac{1}{2^r}\left\lfloor \frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n \right\rfloor \leq \delta$$

Hence:

$$\frac{1}{2^r}\left(\frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n - 1\right) \leq \delta$$
$$\frac{1}{q}\cdot(1-\varepsilon)\cdot 2^n \leq 2^r\cdot\delta + 1 \leq 2^{r+1}\cdot\delta$$

where the last inequality follows from $\delta \geq \frac{1}{2^r}$. After taking logarithms and rearranging:

$$r + 1 \geq n + \log_2(1-\varepsilon) + \log_2\left(\frac{1}{\delta}\right) - \log_2(q)$$

ε is constant, so the claim follows. ■

[2]We assume w.l.o.g. that H always output a value which it has queried.

Corollary 6 *A hitter H with optimal query complexity has* $R_H(\varepsilon, \delta, n) \geq n + \Omega(\log\left(\frac{1}{\delta}\right)) - \log_2\left(\frac{1}{\varepsilon}\right)$

Proof: If we assume the H is optimal with respect to the number of queries it makes, then $Q_H(\varepsilon, \delta, n) = O(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right))$. This gives us:

$$R(H) \geq n + \log_2\left(\frac{1}{\delta}\right) - \log_2(O(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)))$$

And therefore

$$\begin{aligned} R(H) &\geq n + \log_2\left(\frac{1}{\delta}\right) - \log_2\left(\frac{1}{\varepsilon}\right) - \log_2\log\left(\frac{1}{\delta}\right) - O(1) \\ &= n + \Omega(\log\left(\frac{1}{\delta}\right)) - \log_2\left(\frac{1}{\varepsilon}\right) \end{aligned}$$

■

2 Simple Algorithms

In the rest of this lecture, we present several constructions that fulfill the time bound and analyze their query and randomness complexities. This chapter shows three simple constructions.

2.1 Constructing a Hitter from a Boolean Sampler

First, we show that any boolean sampler yields a hitter with related complexity. Using a boolean sampler S, we construct the hitter MONITOR-HITTER$_S$ thus:

Algorithm: MONITOR-HITTER$^f_S(\varepsilon, \delta, n)$
Invoke S^f with parameters $(\frac{\varepsilon}{2}, \frac{\delta}{2}, n)$. Monitor the operation of S^f: if some query x is answered by $f(x) = 1$ then output x and halt. If S halts before the above happens, output $\perp$.

Proposition 2.1 MONITOR-HITTER$_S$ *is a hitter.*

Proof: Let S be a boolean sampler, let $f \in F(n, \varepsilon)$. Let MONITOR-HITTER$^f_S(\varepsilon, \delta, n)$ be a random variable taken from output probability distribution of MONITOR-HITTER^f_S on input (ε, δ, n). We need to prove that

$$\Pr\left[\text{MONITOR-HITTER}^f_S(\varepsilon, \delta, n) \notin f^{-1}(1)\right] < \delta$$

This is exactly the probability that the invoked sampler $S^f\left(\frac{\varepsilon}{2}, \frac{\delta}{2}, n\right)$ does not receive any "1" answers on its queries. Define the random variable $s = S^f\left(\frac{\varepsilon}{2}, \frac{\delta}{2}, n\right)$ to represent the output of S on input $\left(\frac{\varepsilon}{2}, \frac{\delta}{2}, n\right)$ and access to the oracle f. Then:

$$\begin{aligned} &\Pr\left[S^f \text{ receives only 0's}\right] = \\ &\quad \Pr\left[S^f \text{ receives only 0's and } s < \tfrac{\varepsilon}{2}\right] + \Pr\left[S^f \text{ receives only 0's and } s \geq \tfrac{\varepsilon}{2}\right] \end{aligned} \tag{2}$$

Note that $\bar{f} > \varepsilon$ (because $f \in F(n, \varepsilon)$), and since S is a sampler we have:

$$\Pr\left[s < \frac{\varepsilon}{2}\right] < \frac{\delta}{2} \tag{3}$$

On the other hand: in all cases where the coin tosses of S caused S to receive only 0's, they will cause exactly the same scenario if f were replaced by the constant zero function f_0 ($\forall x : f_0(x) = 0$), for which $\bar{f}_0 = 0$. Thus:

$$\Pr\left[s \geq \frac{\varepsilon}{2} \text{ and S receives only 0's}\right] \leq \Pr\left[S^{f_0}\left(\frac{\varepsilon}{2}, \frac{\delta}{2}, n\right) \geq \frac{\varepsilon}{2}\right] < \frac{\delta}{2} \tag{4}$$

Combining equations (2), (3) and (4) we get:

$$\Pr\left[\textsc{Monitor-Hitter}^f_S(\varepsilon, \delta, n) \notin f^{-1}(1)\right] = \Pr\left[S^f \text{ receives only 0's}\right] < \delta$$

■

Note that using the lower bound $Q_H \geq \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$ on the query complexity of hitters, this reduction immediately yields a (non-tight) lower bound of $Q_S \geq \frac{1}{2\varepsilon}\log\left(\frac{1}{2\delta}\right)$ on the query complexity of samplers. Similarly, the $R_H \geq n + \log_2\left(\frac{1}{\delta}\right) - \log_2(Q_H(\varepsilon, \delta, n)) - O(1)$ lower bound on the randomness complexity of hitters implies a lower bound $R_S \geq n + \log_2\left(\frac{1}{2\delta}\right) - \log_2(Q_S(2\varepsilon, 2\delta, n)) - O(1)$ on the randomness complexity of samplers.

2.2 A Naive Hitter

The simplest method to obtain a hitter is to sample random points uniformly, and independently, hoping to hit a good one:

Algorithm: $\textsc{Naive-Hitter}^f(\varepsilon, \delta, n)$

Set $q = \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$.
Randomly choose $x_1, \ldots, x_q \in \{0,1\}^n$, uniformly and independently.
Query f on $x_1, \ldots, x_q$.
If $f(x_i) = 1$ for some i then output the first such x_i. Otherwise output 0.

Proposition 2.2 NAIVE-HITTER *is a hitter.*

Proof: For any $f \in F(n, \varepsilon)$,

$$\Pr\left[\textsc{Naive-Hitter}^f(n, \varepsilon, \delta) \in f^{-1}(1)\right] = 1 - \Pr\left[\forall i, f(x_i) = 0\right] \geq 1 - (1-\varepsilon)^q = 1 - \delta$$

■

The complexities of NAIVE-HITTER are:

- $Q_{\textsc{Naive-Hitter}}(\delta, \varepsilon, n) = q = \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$ — this matches the lower bound
- $R_{\textsc{Naive-Hitter}}(\delta, \varepsilon, n) = q \cdot n = \frac{n}{\varepsilon}\log\left(\frac{1}{\delta}\right)$

We see that by simply choosing an appropriate number of samples at random, we attain optimal (up to a constant multiplicative factor) query complexity. However, the randomness complexity is much higher than the lower bound.

2.3 A Naive Sampler

A similar approach yields a sampler (and hence a boolean sampler):

Algorithm: $\text{NAIVE-SAMPLER}^f(\varepsilon, \delta, n)$

Set $q = \frac{1}{\varepsilon^2} \log\left(\frac{2}{\delta}\right)$.
Randomly choose $x_1, \ldots, x_q \in \{0,1\}^n$, uniformly and independently.
Query f on $x_1, \ldots, x_q$.
Output $\frac{1}{q} \sum_i f(x_i)$.

Proposition 2.3 NAIVE-SAMPLER *is a sampler.*

Proof: Observe that for each x_i, the expected value of $f(x_i)$ is $\bar{f}$. Using Chernoff's bound:

$$\Pr\left[\left|\text{NAIVE-SAMPLER}^f(n, \varepsilon, \delta) - \bar{f}\right| > \varepsilon\right] =$$
$$\Pr\left[\left|\frac{1}{q}\sum_i f(x_i) - \frac{1}{q}\sum_i \mathsf{E}(f(x_i))\right| > \varepsilon\right] < 2e^{-\varepsilon^2 q} = \delta$$

■

The complexities of NAIVE-SAMPLER are:

- $Q_{\text{NAIVE-SAMPLER}}(\varepsilon, \delta, n) = q = \frac{1}{\varepsilon^2} \log\left(\frac{2}{\delta}\right)$ — this matches the lower bound
- $R_{\text{NAIVE-SAMPLER}}(\varepsilon, \delta, n) = n \cdot q = \frac{n}{\varepsilon^2} \log\left(\frac{2}{\delta}\right)$

Similarly to NAIVE-HITTER, the query complexity is optimal (up to a constant multiplicative factor) but the randomness complexity is bad.

3 Better Constructions

We have seen that the naive approach requires too much randomness. In previous lectures we have seen approaches for generating objects that are "close to random" in various senses while using few random bits. Using these approaches we can build samplers and hitters with reduced complexity.

3.1 A Pairwise-Independent Hitter

First we consider *small pairwise independent samples spaces* (Lectures 3 and 4). For concreteness, consider the construction based on finite fields. We wish to produce pairwise-independent elements from $\{0,1\}^n$, so we can use the field $GF(2^n) \equiv \mathsf{Z}_2[x]/f(x)$ where $f(x)$ is an irreducible polynomial in $\mathsf{Z}_2[x]$. Assuming we need $q \leq 2^n$ elements, the construction yields a sample space of size merely $2n$. That is, only $2n$ bits of randomness are required to get q pairwise-independent n-bit elements.

Algorithm: $\text{PAIRWISE-HITTER}^f(\varepsilon, \delta, n)$

Set $q = \frac{1}{\varepsilon \cdot \delta}$.
If $q > 2^n$ then evaluate $f(x)$ for all $x \in \{0,1\}^n$ and output the first $x \in f^{-1}(1)$. Otherwise:
Generate a sequence of q pairwise independent elements $x_1, \ldots, x_q \in \{0,1\}^n$.
Query f on $x_1, \ldots, x_q$.
If for some i, $f(x_i) = 1$, output x_i. Otherwise output $\perp$.

Proposition 3.1 PAIRWISE-HITTER *is a hitter.*

Proof: For $q > 2^n$, correctness is obvious. Otherwise, assume $f \in F(n, \varepsilon)$ and define q random variables $\zeta_1, \ldots, \zeta_q$, where

$$\zeta_i = \begin{cases} 1 & \text{if } f(x_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

We have that $\mathsf{E}(\zeta_i) = \bar{f}$ and $\mathrm{Var}(\zeta_i) = \mathsf{E}({\zeta_i}^2) - \mathsf{E}(\zeta_i)^2 = \mathsf{E}(\zeta_i) - \mathsf{E}(\zeta_i)^2 = \bar{f} \cdot (1 - \bar{f})$. Using Chebychev's inequality:

$$\begin{aligned}
\Pr\left[\text{PAIRWISE-HITTER} \notin f^{-1}(1)\right] &= \Pr\left[\tfrac{1}{q}\sum_i \zeta_i = 0\right] \\
&\leq \Pr\left[\left|\tfrac{1}{q}\sum_i \zeta_i - \bar{f}\right| > \bar{f}\right] \\
&< \frac{\mathrm{Var}(\zeta_i)}{q \cdot \bar{f}^2} = \frac{\bar{f} \cdot (1 - \bar{f})}{q \cdot \bar{f}^2} \\
&< \frac{\bar{f}}{q \cdot \bar{f}^2} = \tfrac{1}{q \cdot \bar{f}} \leq \tfrac{1}{q \cdot \varepsilon} = \delta
\end{aligned}$$

■

The complexities of PAIRWISE-HITTER are:

- $Q_{\text{PAIRWISE-HITTER}}(\varepsilon, \delta, n) \leq q = \frac{1}{\varepsilon \cdot \delta}$
- $R_{\text{PAIRWISE-HITTER}}(\varepsilon, \delta, n) = 2n$

Note that this randomness complexity is optimal (up to a constant factor) for any value of ε. The query complexity is optimal when δ is constant, but otherwise it is not optimal.

3.2 A Pairwise-Independent Sampler

Similarly, we obtain a sampler based on pairwise-independent elements of $\{0, 1\}^n$.

Algorithm: PAIRWISE-SAMPLER$^f(\varepsilon, \delta, n)$

Set $q = \frac{1}{\varepsilon^2 \cdot \delta}$.
If $q > 2^n$ then evaluate $f(x)$ for all $x \in \{0, 1\}^n$ and output the average. Otherwise:
Generate a sequence of q pairwise independent elements $x_1, \ldots, x_q \in \{0, 1\}^n$ using the finite-field construction.
Query f on $x_1, \ldots, x_q$.
Output $\frac{1}{q}\sum_i f(x_i)$.

Proposition 3.2 PAIRWISE-SAMPLER *is a sampler.*

Proof: For $q > 2^n$ correctness is obvious. Otherwise, define q random variables $\zeta_1, \ldots, \zeta_q$, where

$$\zeta_i = \begin{cases} 1 & \text{if } f(x_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

As before, $\mathsf{E}(\zeta_i) = \bar{f}$ and $\mathrm{Var}(\zeta_i) = \bar{f}(1-\bar{f})$. Using Chebychev's inequality:

$$\begin{aligned}
\Pr\left[\left|S^f(n,\varepsilon,\delta) - \bar{f}\right| > \varepsilon\right] &= \Pr\left[\left|\frac{1}{q}\sum_i \zeta_i - \bar{f}\right| > \varepsilon\right] \\
&< \frac{\mathrm{Var}(\zeta_i)}{q\cdot\varepsilon^2} = \frac{\bar{f}\cdot(1-\bar{f})}{q\cdot\varepsilon^2} \\
&< \tfrac{1}{q\cdot\varepsilon^2} = \delta
\end{aligned}$$

■

The complexities of PAIRWISE-SAMPLER are:

- $Q_{\text{PAIRWISE-SAMPLER}}(\varepsilon,\delta,n) = q = \frac{1}{\varepsilon^2\cdot\delta}$
- $R_{\text{PAIRWISE-SAMPLER}} = 2n$.

As before, this randomness complexity is optimal (up to a constant factor), whereas the query complexity is optimal only when δ is constant.

3.3 An Expander-Walk Hitter

Another approach to producing "random-looking" elements using little randomness is *random walks on expander graphs* (Lectures 9 and 10). For any given ε, we will need a strongly-constructible family of expander graphs $\{G_n\}$ with algebraic expansion coefficient $\lambda \leq \frac{\varepsilon}{10}$ and degree $d = \mathrm{poly}\left(\frac{1}{\lambda}\right)$.

This can be obtained directly using the second construction mentioned in Lecture 10 . Alternatively, we can start with any construction that has $d = \mathrm{poly}\left(\frac{1}{\lambda}\right)$ and use amplification to achieve sufficiently low λ (i.e., connect each vertex in G_n to all nodes in its k-neighbourhood, where a suitable value of k is chosen using Theorem 3 in Lecture 10 and Proposition 3.1 in Lecture 9). The relation $d = \mathrm{poly}\left(\frac{1}{\lambda}\right)$ is maintained by the amplification.

Algorithm: WALK-HITTER$^f(\varepsilon,\delta,n)$

Choose a strongly-constructible expander graph G_{2^n} as specified above ($\lambda \leq \frac{\varepsilon}{10}$, $d = \mathrm{poly}\left(\frac{1}{\lambda}\right)$).
Set $q = O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$ (the constant is determined by the proof, see below).
Generate a sequence of q elements $x_1,\dots x_q \in \{0,1\}^n$ by conducting a random walk of length q on the expander from a random initial point.
Query f on $x_1,\dots,x_q$. If for some i, $f(x_i)=1$, output the first such x_i. Otherwise output 0.

Proposition 3.3 WALK-HITTER *is a hitter.*

Proof: Assume $f \in F(n,\varepsilon)$. Denote $B = f^{-1}(0)$.

$$\begin{aligned}
&\Pr\left[\text{WALK-HITTER} \notin f^{-1}(1)\right] \\
&\quad = \Pr\left[\text{a walk of length } q \text{ stays in } B\right] \qquad \text{(using Theorem 6 in Lecture 10)} \\
&\quad < \sqrt{\rho(B)}\cdot\left(\sqrt{\rho(B)}+\lambda\right)^{q-1} \\
&\quad \leq \sqrt{1-\varepsilon}\cdot\left(\sqrt{1-\varepsilon}+\lambda\right)^{q-1} \\
&\quad < \left(1-\tfrac{1}{3}\varepsilon\right)\cdot\left(1-\tfrac{1}{3}\varepsilon+\tfrac{1}{10}\varepsilon\right)^{q-1} \\
&\quad \leq \left(1-\tfrac{\varepsilon}{5}\right)^q = \delta
\end{aligned}$$

■

The complexities of WALK-HITTER are:

- $Q_{\text{WALK-HITTER}}(\varepsilon, \delta, n) = q = O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\right)$
- $R_{\text{WALK-HITTER}}(\varepsilon, \delta, n) = n + (q-1)\log d = n + O\left(\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

Note that the query complexity is optimal (up to constant factor), but the randomness complexity is optimal only when ε is constant. This is the reverse of the case for the pairwise-independent constructions above.

4 Composed Constructions

We have seen:

	Q_M	R_M
PAIRWISE-HITTER PAIRWISE-SAMPLER	optimal for constant δ	optimal
WALK-HITTER	optimal	optimal for constant ε

In this section we combine the two constructions, using the "well-handled" parameter (ε or δ) of one to fix the "badly-handled" parameter of the other. We thus obtain composite constructions that are optimal in all parameters.

4.1 A Composed Hitter

PAIRWISE-HITTER uses $2n$ bits of randomness. Consider a deterministic version D-PAIRWISE-HITTER of it, which accepts these $2n$ bits as an additional input. That is,

$$\text{PAIRWISE-HITTER}^f(\varepsilon, \delta, n) \equiv \left[r \in_R \{0,1\}^{2n} \text{ ; return D-PAIRWISE-HITTER}^f(\varepsilon, \delta, n, r)\right]$$

We fix the input δ of D-PAIRWISE-HITTER as $\frac{1}{2}$, since it is not optimal in this parameter. This gives a natural definition of good values of r:

Definition 7 (Good Seed) *$r \in \{0,1\}^{2n}$ is called a* good seed *(w.r.t. f, ε, n) if D-PAIRWISE-HITTER$^f(\varepsilon, \frac{1}{2}, n, r) \in f^{-1}(1)$. Denote $G \stackrel{\text{def}}{=} \{r \mid r \text{ is a good seed}\}$.*

If we come up with a good seed $r \in G$ then we can find some $x \in f^{-1}(1)$ just by running D-PAIRWISE-HITTER$(\varepsilon, \frac{1}{2}, n, r)$ — this costs $\frac{2}{\varepsilon}$ queries and no random bits. Thus we have a new problem: finding elements of G inside $\{0,1\}^{2n}$. For $f \in F(n, \varepsilon)$, PAIRWISE-HITTER$(\varepsilon, \frac{1}{2}, n, r)$ should succeed with probability at least $\frac{1}{2}$ (since we fixed its δ at $\frac{1}{2}$), so at least half the choices of r are good seeds: $|G| \geq \frac{1}{2}2^{2n}$. Hence the problem of finding good seeds is just the job for a hitter with $\varepsilon = \frac{1}{2}$. WALK-HITTER is an attractive choice, since it is optimal for such fixed ε. This yields the following algorithm.

Algorithm: $\textsc{Composed-Hitter}^f(\varepsilon, \delta, n)$

Define the following deterministic subroutine (using the oracle f and the inputs ε and n):

$$\textsc{Check-Seed}(r) \equiv \left[\text{return } f\left(\textsc{d-Pairwise-Hitter}^f(\varepsilon, \tfrac{1}{2}, n, r)\right)\right] \qquad \text{where } r \in \{0,1\}^{2n}$$

Compute $r \leftarrow \textsc{Walk-Hitter}^{\textsc{Check-Seed}}(2n, \frac{1}{2}, \delta)$; that is, execute $\textsc{Walk-Hitter}$ with input $(2n, \frac{1}{2}, \delta)$ and answer its oracle queries using $\textsc{Check-Seed}$.
Output $\textsc{d-Pairwise-Hitter}(\varepsilon, \frac{1}{2}, n, r)$.

Proposition 4.1 $\textsc{Composed-Hitter}$ *is a hitter.*

Proof: Note that by definition r is a good seed iff $\textsc{Check-Seed}(r) = 1$. Since $f \in F(n, \varepsilon)$ and $\textsc{Pairwise-Hitter}$ is a sampler, at least half of the r values are good, i.e., $\textsc{Check-Seed} \in F(2n, \frac{1}{2})$. Thus by $\textsc{Walk-Hitter}$ being a hitter:

$$\Pr\left[\textsc{Walk-Hitter}^{\textsc{Check-Seed}}(\tfrac{1}{2}, \delta, 2n) \in G\right] > 1 - \delta$$

Hence with probability at least $1 - \delta$, $\textsc{Walk-Hitter}$ outputs a good seed r, and in this case $\textsc{d-Pairwise-Hitter}(\varepsilon, \frac{1}{2}, n, r)$ outputs x such that $f(x) = 1$. ■

The complexities of $\textsc{Composed-Hitter}$ are:

- $Q_{\textsc{Composed-Hitter}}(\varepsilon, \delta, n) \le Q_{\textsc{Walk-Hitter}}(\frac{1}{2}, \delta, 2n) \cdot Q_{\textsc{Pairwise-Hitter}}(\varepsilon, \frac{1}{2}, n) = O\left(2 \cdot \log\left(\frac{1}{\delta}\right)\right) \cdot \left(\frac{2}{\varepsilon}\right) = O\left(\frac{1}{\varepsilon}\right) \cdot \log\left(\frac{1}{\delta}\right)$
- $R_{\textsc{Composed-Hitter}}(\varepsilon, \delta, n) = R_{\textsc{Walk-Hitter}}(\frac{1}{2}, \delta, 2n) = 2n + O\left(\log\left(\frac{1}{\delta}\right)\right)$

Both complexities are optimal, since they match the corresponding lower bounds (up to a constant factor).

4.2 A Composed Sampler

We have obtained an optimal hitter by comining two suboptimal hitters. Can we do the same for samplers? There are some difficulties: first, we haven't seen any sampler with the characteristics of $\textsc{Walk-Hitter}$ (namely, optimal query complexity for constant ε). Furthermore, in the two-level construction we have seen, the first level needs to somehow combine the results of the second level. For hitters this was trivial, but here it's not clear whether using a sampler in the first level to estimate the average of the second one would work. By appropriate adaptation of the two-level scheme we indeed obtain an optimal sampler.

As before, we use a construction based on pairwise-independent sample spaces for the second level. $\textsc{Pairwise-Sampler}$ uses $2n$ bits of randomness, so we consider a deterministic version $\textsc{d-Pairwise-Sampler}$ of it which accepts these $2n$ bits as an additional input. That is,

$$\textsc{Pairwise-Sampler}^f(\varepsilon, \delta, n) \equiv \left[r \in_R \{0,1\}^{2n} \text{ ; return } \textsc{d-Pairwise-Sampler}^f(\varepsilon, \delta, n, r)\right]$$

Definition 8 (Good Seed) $r \in \{0,1\}^{2n}$ *is called a* good seed *(w.r.t. f, ε, n) if* $\left|\textsc{d-Pairwise-Sampler}^f(\varepsilon, \frac{1}{100}, n, r) - \overline{f}\right| < \varepsilon$. *Denote* $G \stackrel{\text{def}}{=} \{r \mid r \text{ is a good seed}\}$.

Unlike the case for hitters, we can't easily know whether a seed is good. However, we know that *most* seeds are good, and we can use random walk on an expander graph to obtain a set of seeds in which, with high probability, most seeds are good. This intuition yields the following algorithm.

Algorithm: COMPOSED-SAMPLER$^f(\varepsilon, \delta, n)$

Choose a strongly-constructible expander graph $G_{2^{2n}}$ with an algebraic expansion coefficient $\lambda \leq \frac{1}{100}$ and degree $d = \text{poly}\left(\frac{1}{\lambda}\right) = \frac{1}{O(1)}$ (the constants are fixed in the proof below).
Set $q = O\left(\log\left(\frac{1}{\delta}\right)\right)$
Conduct a random walk of length q on $G_{2^{2n}}$ to generate a sequence of q seeds $r_1, \ldots, r_q \in \{0,1\}^{2n}$.
For each $i \in \{1, \ldots, q\}$, compute $s_i =$ D-PAIRWISE-SAMPLER$(\varepsilon, \frac{1}{100}, n, r_i)$.
Output the median of $\{s_1, \ldots, s_q\}$.

To prove correctness we need the following generalization of Theorem 6 in Lecture 10:

Theorem 9 *Let $G = (V, E)$ be an expander with algebraic expansion coefficient λ. Let $B_1, \ldots B_l \subseteq V$. Let the random variables $v_1, \ldots, v_l$ denote the vertices encountered in a random walk of length l on G with a random starting point. For any $B \subseteq V$ denote $\rho(B) = \frac{|B|}{|V|}$. Then:*

$$\Pr\left[\forall i : v_i \in B_i\right] \leq \sqrt{\rho\left(B_1\right)} \prod_{i=2}^{l} \left(\sqrt{\rho\left(B_i\right)} + \lambda\right)$$

Proof: The proof of this theorem is very similar to the proof of Theorem 6 in Lecture 10. For a given set $B \subseteq V$, define the projection:

$$P_B\left(x\right) = \begin{cases} 1 & \text{if } x \in B \\ 0 & \text{otherwise} \end{cases}$$

We define π to be the vector in R^V for which $\forall v \in V : \pi_v = \frac{1}{n}$ (i.e., π represents the uniform distribution on V). The following holds (similarly to the proof of Theorem 6 in Lecture 10):

$$\Pr\left[\forall i : v_i \in B_i\right] = \left\|P_{B_l} A P_{B_{l-1}} A \ldots P_{B_1} \pi\right\|_2$$

Recall the following lemma:

Lemma 4.2 (Lemma 3.1 in Lecture 10) *For every set $B \subseteq V$ and vector $\alpha \in R^V$:*

$$\left\|P_B A \alpha\right\|_2 \leq \left(\sqrt{\rho\left(B\right)} + \lambda\right) \cdot \|\alpha\|_2$$

Denote $N = |V|$. By l successive applications of Lemma 4.2, we obtain:

$$\begin{aligned} \|P_{B_l} A P_{B_{l-1}} A \ldots P_{B_1} \pi\|_1 &\leq \sqrt{N} \cdot \left\|P_{B_l} A P_{B_{l-1}} A \ldots P_{B_1} \pi\right\|_2 \\ &\leq \sqrt{N} \cdot \textstyle\prod_{i=2}^{l} \left(\sqrt{\rho(B_l)} + \lambda\right) \|P_{B_1} \pi\|_2 \\ &= \sqrt{N} \cdot \textstyle\prod_{i=2}^{l} \left(\sqrt{\rho(B_l)} + \lambda\right) \sqrt{\frac{\rho(B_1)}{N}} \\ &= \sqrt{\rho\left(B_1\right)} \textstyle\prod_{i=2}^{l} \left(\sqrt{\rho\left(B_i\right)} + \lambda\right) \end{aligned}$$

■

Using Theorem 9, we prove the following:

Lemma 4.3 *Let $G = (V, E)$ be an expander with algebraic expansion coefficient λ. Let $B \subseteq V$ and denote $\rho(B) = \frac{|B|}{|V|}$. Let the random variables $v_1, \ldots, v_l$ denote the vertices encountered in a random walk of length l on G with a random starting point. Then:*

$$\Pr\left[|\{i : v_i \in B\}| \geq \tfrac{1}{2}l\right] \leq \left(8\sqrt{\rho(B)} + 8\lambda\right)^{\frac{1}{2}l}$$

Proof: For a given $I \subseteq \{1, \ldots, l\}$, we set:

$$B_i^I = \begin{cases} B & \text{if } i \in I \\ V & \text{otherwise} \end{cases}$$

Using Theorem 9:

$$\begin{aligned}
\Pr\left[|\{i : v_i \in B\}| \geq \tfrac{1}{2}l\right] &\leq \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \Pr\left[\forall i \in V : v_i \in B_i^I\right] \\
&< \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \sqrt{\rho(B_1^I)} \prod_{i=2}^{l} \left(\sqrt{\rho(B_i^I)} + \lambda\right) \\
&< \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \prod_{i=1}^{l} \left(\sqrt{\rho(B_i^I)} + \lambda\right) \\
&< \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \left(\prod_{i \in I} \left(\sqrt{\rho(B_i^I)} + \lambda\right) \cdot \prod_{i \notin I} \left(\sqrt{\rho(B_i^I)} + \lambda\right)\right) \\
&= \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \left(\prod_{i \in I} \left(\sqrt{\rho(B)} + \lambda\right) \cdot \prod_{i \notin I} (1 + \lambda)\right) \\
&= \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \left(\left(\sqrt{\rho(B)} + \lambda\right)^{|I|} \cdot \left(1 + \lambda\right)^{l-|I|}\right) \\
&\leq \sum_{I \subseteq \{1,\ldots,l\},\, |I| \geq \frac{1}{2}l} \left(\left(\sqrt{\rho(B)} + \lambda\right)^{\frac{1}{2}l} \cdot \left(1 + \lambda\right)^{\frac{1}{2}l}\right) \\
&\leq 2^l \cdot \left(1 + \lambda\right)^{\frac{1}{2}l} \cdot \left(\sqrt{\rho(B)} + \lambda\right)^{\frac{1}{2}l}
\end{aligned}$$

Since $\lambda < 1$, we obtain:

$$\begin{aligned}
\Pr\left[|\{i : v_i \in B\}| \leq \tfrac{1}{2}l\right] &< 2^{\frac{3}{2}l} \left(\sqrt{\rho(B)} + \lambda\right)^{\frac{1}{2}l} \\
&= \left(8\sqrt{\rho(B)} + 8\lambda\right)^{\frac{1}{2}l}
\end{aligned}$$

■

To prove correctness of our sampler we also need the following technical fact:

Fact 4.4 *Let $A = \{a_1, \ldots, a_n\}$ be a set of real numbers and $x < y$ be also real numbers. If there exists a subset $I \subseteq A$ s.t. $|I| > \frac{1}{2}n$ and all elements in I fall in the interval (a, b) then the median of A also falls in this interval. In other words, if $|\{a \in A : x < a < y\}| > \frac{1}{2}n$, then $x < \mathrm{median}(A) < y$.*

Proof: Define:

$$\begin{aligned} G &= \{a \in A \mid a \geq \mathrm{median}(A)\} \\ L &= \{a \in A \mid a \leq \mathrm{median}(A)\} \end{aligned}$$

By the definition of a median: $|G| \geq \lceil \frac{1}{2}n \rceil$ and $|L| \geq \lceil \frac{1}{2}n \rceil$. Since $|I| > \frac{1}{2}n$, there exist some $g \in G \cap I$ and $l \in L \cap I$. Because $g \in I$ we have $g < y$, so using $g \in G$ we get $\mathrm{median}(A) \leq g < y$. Similarly, because $l \in I$ we have $x < l$, so using $l \in L$ we get $\mathrm{median}(A) \geq l > x$. The claim follows. ■

We can now prove correctness of the sampler.

Proposition 4.5 COMPOSED-SAMPLER *is a sampler.*

Proof: Consider the execution of $\text{COMPOSED-SAMPLER}^f(\varepsilon, \delta, n)$ with some $f : \{0,1\}^n \to \{0,1\}$.

Recall that the algorithm produces q seeds $\{r_1, \ldots, 1_q\}$ and runs D-PAIRWISE-SAMPLER on each of them to obtain corresonding values $\{s_1, \ldots, r_q\}$. For each seed r_i, if r_i is a good seed then by definition: $|s_i - \bar{f}| < \varepsilon$. If strictly more than one half of $\{r_1, \ldots, r_q\}$ are good seeds then by Fact 4.4, the output $s = \mathrm{median}\{s_1, \ldots, s_q\}$ also satisfies $|s - \bar{f}| < \varepsilon$. Thus, it suffices to show that with probability at least δ, strictly more than one half of $\{r_1, \ldots, r_q\}$ are good seeds.

For every ε, n and $f \in F(n)$, at least $\frac{99}{100}$ of the seeds are good (by PAIRWISE-SAMPLER$(\varepsilon, \frac{1}{100}, n)$ being a sampler), i.e., $|G| \geq \frac{99}{100}|\{0,1\}^{2n}|$. Letting $B = \{0,1\}^{2n} \setminus G$ be the set of "bad" seeds, we obtain $\rho(B) \leq \frac{1}{100}$. Thus, by invoking Lemma 4.3 on G_{2n} we obtain:

$$\begin{aligned} \Pr[\text{more than one half of } \{r_1, \ldots, r_q\} \text{ are good seeds}] &= 1 - \Pr\left[|\{i : r_i \in B\}| > \tfrac{1}{2}q\right] \\ &\leq \left(8\sqrt{\rho(B)} + 8\lambda\right)^{\frac{1}{2}q} \\ &\leq \left(8 \cdot \sqrt{\frac{1}{100}} + 8 \cdot \frac{1}{100}\right)^{\frac{1}{2}q} \\ &\leq 0.88^{\frac{1}{2}q} = \delta \end{aligned}$$

q is chosen as to satisfy the last inequality. ■

The complexities of COMPOSED-SAMPLER are as follow (where $q = O\left(\log\left(\frac{1}{\delta}\right)\right)$ is the number of tested seeds):

- $Q_{\text{COMPOSED-SAMPLER}}(\varepsilon, \delta, n) = q \cdot Q_{\text{PAIRWISE-SAMPLER}}(n, \varepsilon, \frac{1}{100})) = O\left(\log\left(\frac{1}{\delta}\right)\right) \cdot \frac{1}{\varepsilon^2 \cdot 100} = O\left(\log\left(\frac{1}{\delta}\right) \cdot \frac{1}{\varepsilon^2}\right)$

- $R_{\text{COMPOSED-SAMPLER}}(\varepsilon, \delta, n) = 2n + O\left(\log\left(\frac{1}{\delta}\right) \cdot \log(d)\right) = 2n + O\left(\log\left(\frac{1}{\delta}\right)\right)$

This result is optimal, since it matches the lower bounds (up to a constant factor).

References

[1] M. Bellare, O. Goldreich and S. Goldwasser, *Randomness in Interactive Proofs*, Computational Complexity, vol. 4, no. 1, pp. 319–354, 1993.
Preliminary version in Proc. of 31st Symposium on Foundations of Computer Science, pp. 563–571, 1990

[2] R. Canetti, G. Even and O. Goldreich, *Lower Bounds for Sampling Algorithms for Estimating the Average*, Information Processing Letters, vol. 53, pp. 17–25, 1995
First pusblished as TR-686, Computer Science Dept., Technion, Haifa, Israel, 1991

[3] A. C. Yao, *Probabilistic computations: Towards a Unified Measure of Complexity*, Proc. 17th Symposium on Foundations of Computer Science, pp. 222-227, 1977