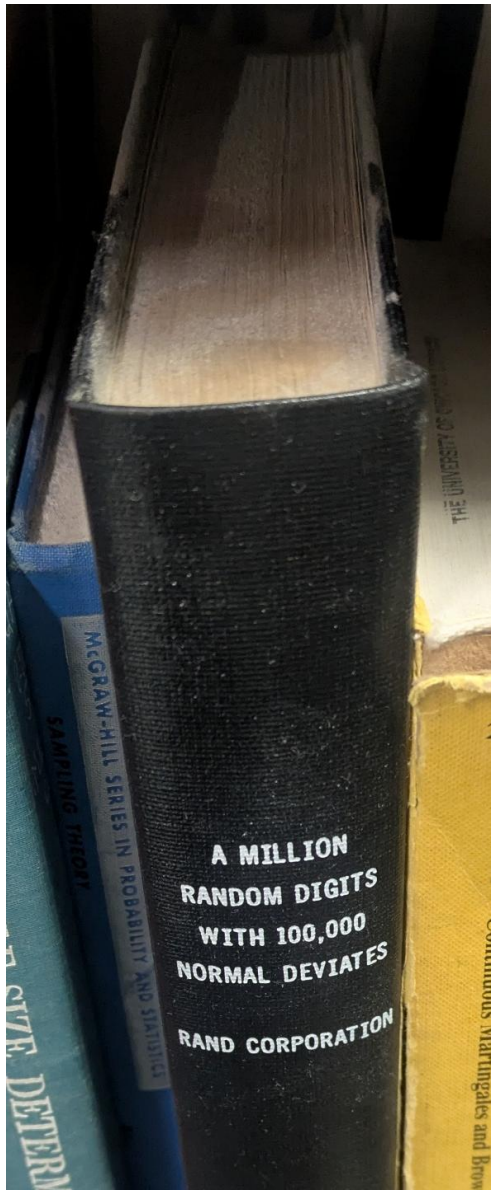
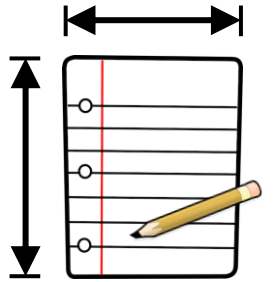


Multi-Access Randomness Saves Space, Even for Halting Algorithms

William Hoza
The University of Chicago

Randomness and space complexity

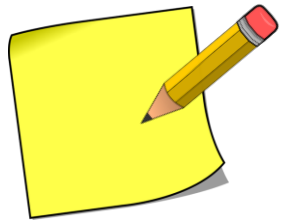


- **Question:** Is randomness useful for log-space computation?
- **Answer:** It depends!
- There are multiple reasonable models of randomized log-space computation

The standard model



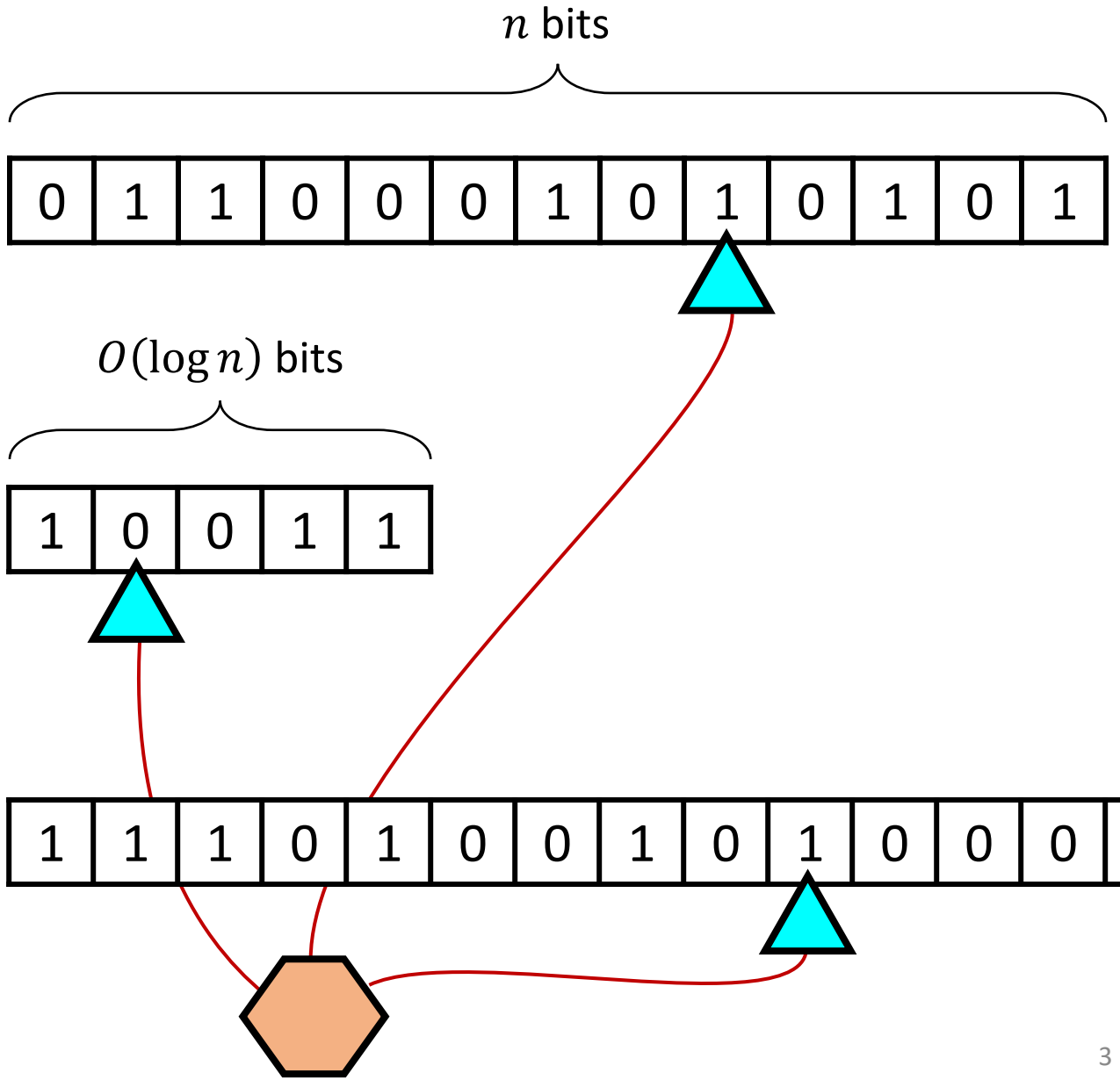
Read-only
input tape



Read-write
work tape

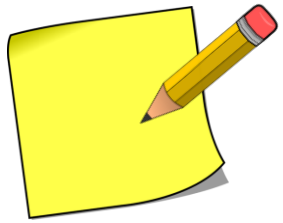


Read-once
random tape



The standard model







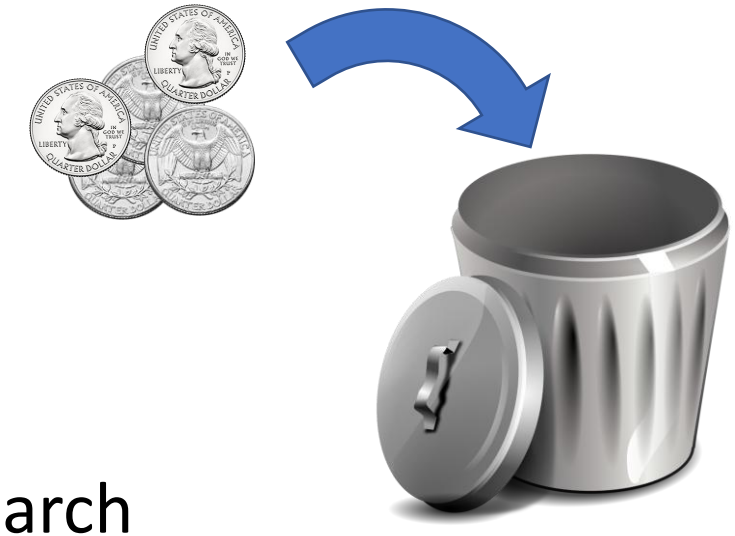
- Let $L: \{0,1\}^* \to \{0,1\}$

Usually denoted BPL

- **Definition:** $L \in \mathrm{BP_H L}$ if there exists a Turing machine M such that:
  - There is a read-only tape storing the n-bit input
  - There is a read/write work tape of size $O(\log n)$
  - There is a read-once tape filled with random bits
  - For every $x \in \{0,1\}^*$, we have $\Pr[M(x) = L(x)] \geq 2/3$
  - M halts for every input and every setting of the random tape

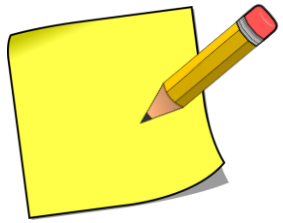
Derandomizing the standard model

Theorem [OpenAI 2026]: $\mathrm{L} = \mathrm{BP_H L}$



- Culmination of several decades of intense research
- **Interpretation:** In the standard model of randomized log-space computation, randomness is not useful

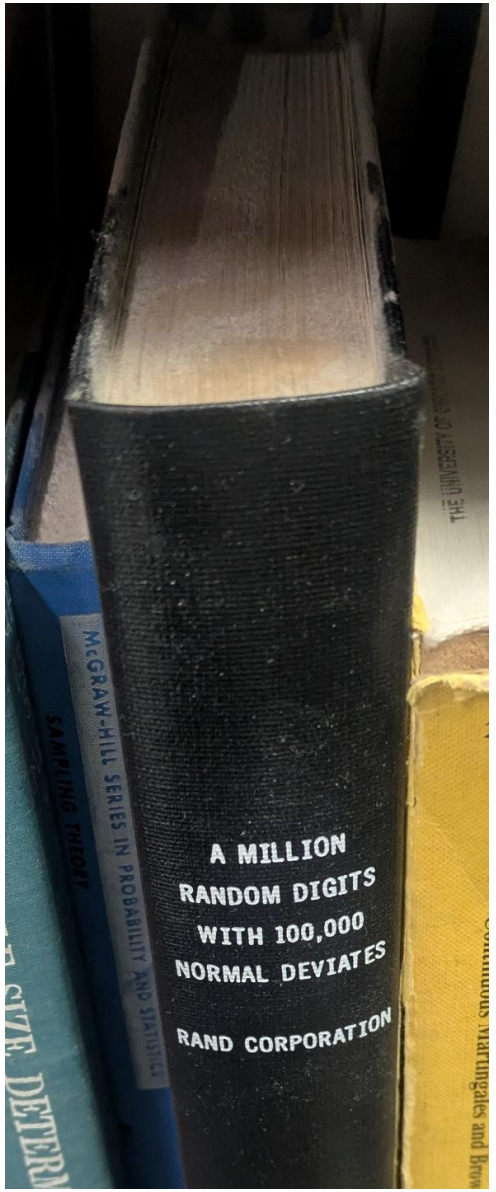
Non-halting model


- Define $\mathrm{BP}_\infty\mathrm{L}$ just like $\mathrm{BP}_\mathrm{H}\mathrm{L}$, except we don't require that the machine always halts
- $\mathrm{NL} \subseteq \mathrm{BP}_\infty\mathrm{L}$ [Gill 1977], so randomness plausibly is useful in this setting!
- **Conjecture** [Saks 1996]: $\mathrm{BP}_\infty\mathrm{L} = \mathrm{NL}$
- $\mathrm{BP}_\infty\mathrm{L} \subseteq \mathrm{TC}^1$ [Borodin, Cook, Pippenger 1983]

Multi-access randomness

- Define $\mathrm{BP}^*_\infty\mathrm{L}$ just like $\mathrm{BP}_\infty\mathrm{L}$, except we give the machine a multi-access random tape (head can move left or right)
- Interpretation:
  - A read-once random tape represents a single coin
  - A multi-access random tape represents a huge database of pre-generated random bits



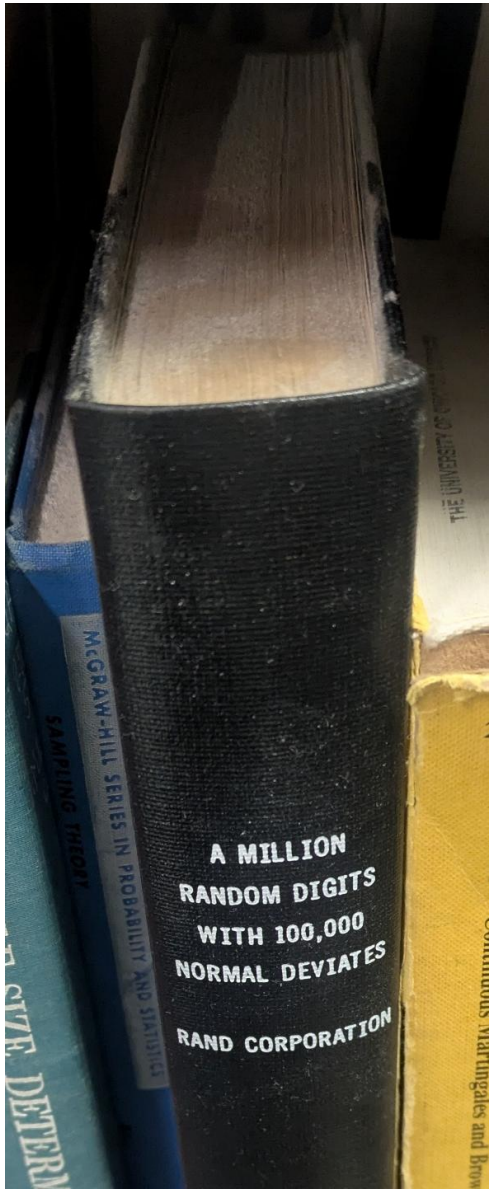
Multi-access randomness

- $\mathrm{BP}_{\infty}^{*}\mathrm{L} = \mathrm{PSPACE}$

  [Karpinsky, Verbeek 1985] [Karpinsky, Verbeek 1986]

- By the Space Hierarchy Theorem, $\mathrm{L} \neq \mathrm{PSPACE}$

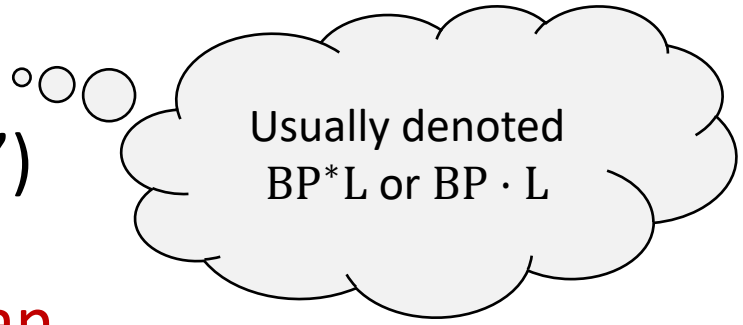
- So randomness is provably useful in the $\mathrm{BP}_{\infty}^{*}\mathrm{L}$ model



Polynomial-time computation



- Define $\mathrm{BP}^*_{\mathrm{T}}\mathrm{L}$ just like $\mathrm{BP}^*_{\infty}\mathrm{L}$, except the machine is required to halt in polynomial time ("T" for "Time")
- $\mathrm{L} = \mathrm{BP}^*_{\mathrm{T}}\mathrm{L}$ under plausible assumptions [Klivans, Van Melkebeek 2002] [Pass, Renard 2023] [Shalunov 2024]



Six models of randomized log-space

	Polynomial time (T)	Finite time (H)	Non-halting (∞)
Read-once randomness	$\mathrm{BP_TL}$	$\mathrm{BP_HL}$	$\mathrm{BP_\infty L}$
Multi-access randomness (*)	$\mathrm{BP_T^*L}$	$\mathrm{BP_H^*L}$	$\mathrm{BP_\infty^*L}$

Six models of randomized log-space

	Polynomial time (T)	Finite time (H)	Non-halting (∞)
Read-once randomness	$\mathrm{BP_TL}$	$\mathrm{BP_HL}$	$\mathrm{BP_\infty L}$
Multi-access randomness (*)	$\mathrm{BP_T^*L}$	$\mathrm{BP_H^*L}$	$\mathrm{BP_\infty^*L}$

Time complexity in read-once model



- **Exercise:** If a log-space machine with a read-once random tape always halts, then it halts in polynomial time
  - Similar: $\mathrm{L} \subseteq \mathrm{P}$
- **Corollary:** $\mathrm{BP_T L} = \mathrm{BP_H L}$
- (Of course, nowadays we know $\mathrm{BP_T L} = \mathrm{BP_H L} = \mathrm{L}$ [OpenAI 2026])

Six models of randomized log-space

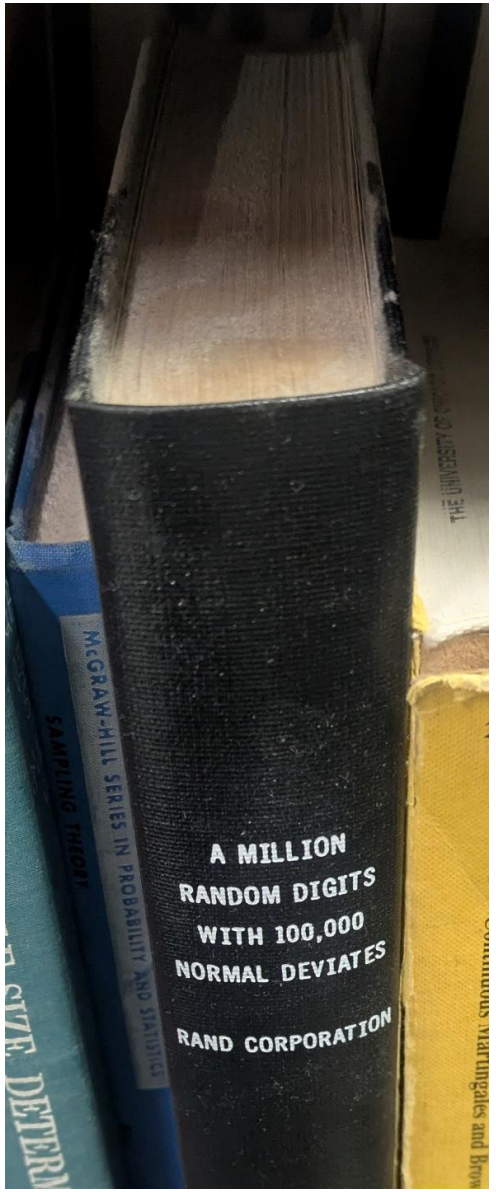
	Polynomial time (T)	Finite time (H)	Non-halting (∞)
Read-once randomness	$\mathrm{BP_TL}$	$\mathrm{BP_HL}$	$\mathrm{BP_\infty L}$
Multi-access randomness (*)	$\mathrm{BP_T^*L}$	$\mathrm{BP_H^*L}$	$\mathrm{BP_\infty^*L}$

Six models of randomized log-space

	Polynomial time (T)	Finite time (H)	Non-halting (∞)
Read-once randomness	$\mathrm{BP_TL}$	$\mathrm{BP_HL}$	$\mathrm{BP_\infty L}$
Multi-access randomness (*)	$\mathrm{BP_T^*L}$	$\mathrm{BP_H^*L}$	$\mathrm{BP_\infty^*L}$

The complexity class $\mathrm{BP}^*_{\mathrm{H}}\mathrm{L}$

- Suppose a log-space machine has a multi-access random tape and always halts
- One might intuitively expect:
  - Such a machine halts in polynomial time [Nisan 1993]
  - $\mathrm{BP}^*_{\mathrm{H}}\mathrm{L} = \mathrm{BP}^*_{\mathrm{T}}\mathrm{L} = \mathrm{L}$



This work: The surprising power of $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$

Main Theorem: $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$

- $\mathrm{P}^{\#\mathrm{P}}$ contains P, NP, PH [Toda 1991], and BQP [Bernstein, Vazirani 1997]
- Our theorem gives strong evidence that $\mathrm{L} \neq \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$
- Our algorithm's time complexity is $2^{\mathrm{poly}(n)}$
  - For comparison, the "$\mathrm{PSPACE} \subseteq \mathrm{BP}_{\infty}^{*}\mathrm{L}$" algorithm has doubly-exponential time complexity [Karpinsky, Verbeek 1985] [Karpinsky, Verbeek 1986]

Proof overview



Lemma 1: $\mathrm{PP} \subseteq \mathrm{BP}^*_{\mathrm{H}}\mathrm{L}$

Lemma 2: $\mathrm{P}^{\mathrm{BP}^*_{\mathrm{H}}\mathrm{L}} = \mathrm{BP}^*_{\mathrm{H}}\mathrm{L}$

- Given the two lemmas, the main theorem will follow, because

$$\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}^{\mathrm{PP}} \subseteq \mathrm{P}^{\mathrm{BP}^*_{\mathrm{H}}\mathrm{L}} = \mathrm{BP}^*_{\mathrm{H}}\mathrm{L}$$

[Gill 1977] Lemma 1 Lemma 2

$\mathrm{SAT} \in \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$ (proof sketch)

- Let ϕ be a given n-variable CNF formula
- **Goal:** Figure out whether ϕ is satisfiable
- **Strategy:** Search the random tape for a satisfying assignment
- **Challenge:** When is it time to give up searching?
  - We don't have enough space to count 2^n trials
  - We must always halt

A probabilistic timer



- **Idea:** After i-th trial, sample a positive integer H_i with $\Pr[H_i = h] = 2^{-h}$
- Halt if we find either:
  1. A large value $H_i \geq c_1 n$, OR
  2. An interval $i, i+1, i+2, \ldots, j$ such that $\max\{H_i, H_{i+1}, \ldots, H_j\}$ occurs at least $c_2 n$ times among $H_i, H_{i+1}, \ldots, H_j$
- **Fact 1:** One of these conditions must happen eventually
  - **Proof sketch:** The only way to avoid condition 2 is if we get larger and larger H_i, eventually satisfying condition 1

A probabilistic timer



- **Idea:** After i-th trial, sample a positive integer H_i with $\Pr[H_i = h] = 2^{-h}$
- Halt if we find either:
  1. A large value $H_i \geq c_1 n$, OR
  2. An interval $i, i+1, i+2, \ldots, j$ such that $\max\{H_i, H_{i+1}, \ldots, H_j\}$ occurs at least $c_2 n$ times among $H_i, H_{i+1}, \ldots, H_j$
- **Fact 2:** With high probability, we do $2^{\Omega(n)}$ trials before halting
  - **Proof sketch:** $\Pr[H_i = h] = \Pr[H_i > h]$, so the probability of getting $c_2 n$ occurrences of h before a single occurrence $> h$ is $2^{-c_2 n}$

Lemma 1: $\mathrm{PP} \subseteq \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$



Lemma 2: $\mathrm{P}^{\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}} = \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$

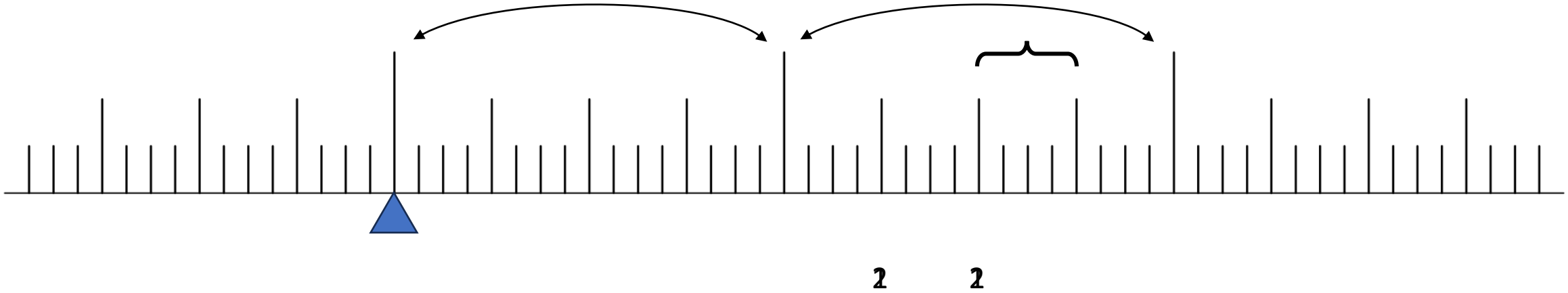
- **Strategy for proving Lemma 2:** We will show how to use the (read-only!) random tape to implement a stack

Implementing a stack

Cf. skip list [Pugh 1990]

Cf. [Küçük, Say, and Yakaryilmaz 2014]

- Sample positive integers $H_1, H_2, \ldots$ with geometrically decaying pmf



- Push(2): Move to subinterval #2
- Do computation in that subinterval
- Pop: Count subintervals

Using the stack to compose algorithms

- Let $f: \{0,1\}^* \to \{0,1\}^*$ and suppose f is length-preserving
- **Claim:** If f can be computed in $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$ model, then $f^{(n)}(x)$ can too
- **Proof sketch:**
    1. Simulate the f algorithm. When it queries its input:
        A. Push its $O(\log n)$-bit configuration onto stack and erase it from work tape
        B. Recursively compute the desired bit of $f^{(n-1)}(x)$
        C. Pop configuration off stack and continue the simulation

Finishing the proof of the lemma

Lemma 2: $\mathrm{P}^{\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}} = \mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$

- **Proof sketch:** For each configuration z of the $\mathrm{P}^{\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}}$ computation, let $f(z)$ be the successor configuration
- Then f can be computed in $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$ model
- By the claim, we can compute $f^{(t)}(z)$, so we can simulate t steps of $\mathrm{P}^{\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}}$ computation

Conclusions

- $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$: A new example of a natural model of computation in which randomness is (apparently) intrinsically useful
- **Open problem:** Determine the exact power of $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L}$
- Best upper bound: $\mathrm{BP}_{\mathrm{H}}^{*}\mathrm{L} \subseteq \mathrm{BP}_{\infty}^{*}\mathrm{L} = \mathrm{PSPACE}$
- **Thanks for watching!**

