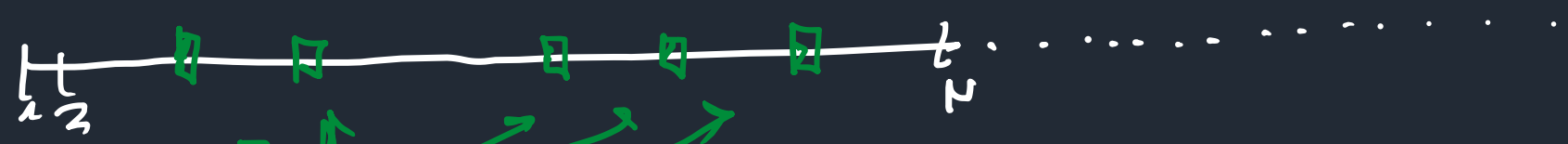
STREAMING : generate random sample



unif. distributed sample

$A \subseteq \{1..N\}$ $|A| = k$ $\quad P[S = A] = \frac{1}{\binom{N}{k}}$.

Basic alg : $k = 1$.

INIT: $p = 0$; data = nil; $n = 0$;

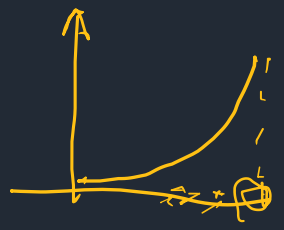
onGet(x) { if (random() $< \frac{1}{n}$) { $p = n$; data = x } n++; }

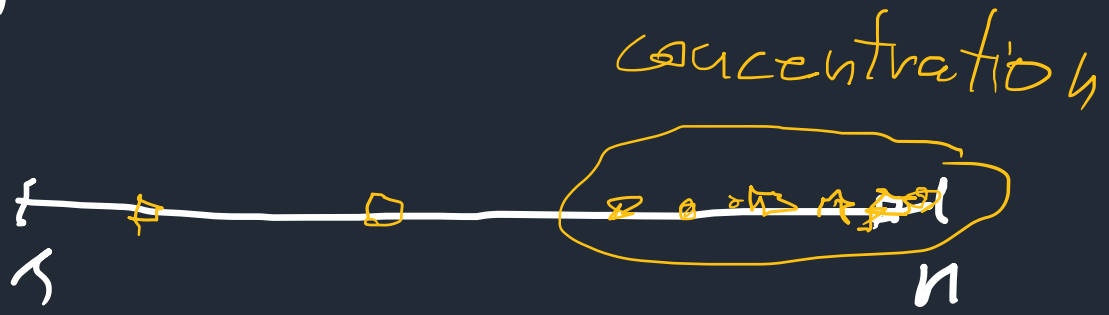
Modifications

```
onGet(x) {
    nt;
    if (random() < 2/n){
        p = n;
        data = x
    }
}
```

$< \frac{1}{n}$

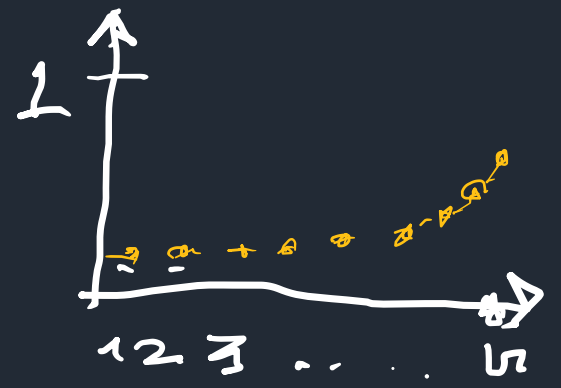
$< \frac{1}{\sqrt{n}}$





$(\text{random}() < \frac{1}{n})$

$(\text{random}() < \frac{2}{n})$



SLIDING WINDOW

$(M = 10^9)$

$N \gg 10^9$

p - rand. variable in $\{1, \dots, M\}$

1) $1 \le p \le M$

2) onGet(x) {
    if ($p = M$) or (rand.dec.(p)) {
        $p = 1$;
    }
    data = x

- 2005: P. Indyk, Motwani
- 2012: Braverman, ...

COUNTING :

- we observe $(a_i)_{i=1..N}$
- calculate (estimate) :

$$|\{a_i : i = 1..N\}|$$

Ex) aabbabbccebbcdcecc = (*)

$$|\{a, b, c, d, e\}| = 5$$

Naive solution : (*) $\xrightarrow[\text{sort}]{}$ aaccccbbbccddee

1 1 1 1 1 2 2 2 3 3 4 4 5

(lg memory)

time : $O(n \log n)$

DO NOT LOOK AT a's

CONSIDER $h(a)$ — h - good hash function

$\{a_1, a_2, a_3, \ldots, a_n\}$

$\downarrow$

$|\{a_1, a_2, a_3, \ldots, a_n\}|$

$\downarrow\uparrow$

$|\{h(a_1), h(a_2), \ldots, h(a_n)\}|$

$i<j : a_i = a_j \rightarrow h(a_i) = h(a_j)$.

<u>suppose</u>: $a \neq b \longrightarrow h(a) \neq h(b)$ with high prob.

collision

LINEAR COUNTING.

We have a table $X[1 \dots N]$ of bits (in memory).

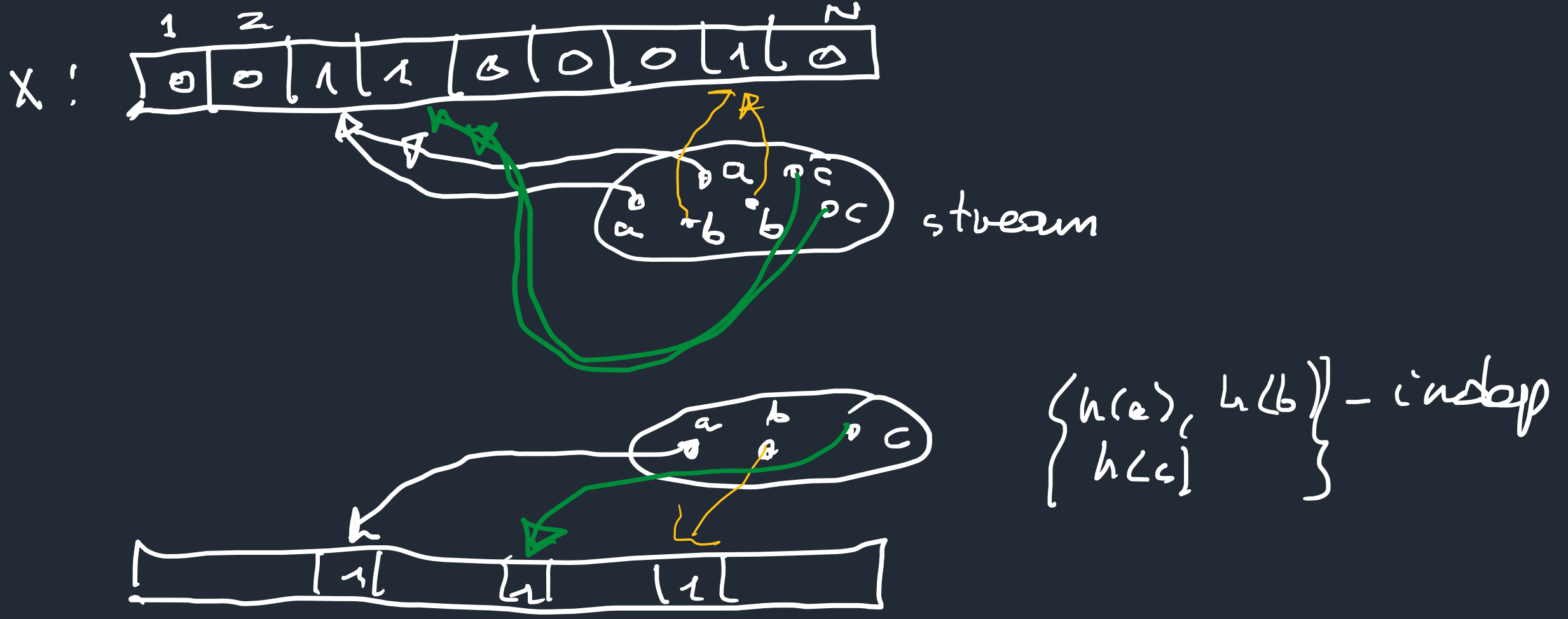
Initially $X = (0, 0, \dots, 0)$

We have a hash function $h : \Omega^* \to \{1, \dots, N\}$

```
onGet(a) {
    i = h(a);
    X[i] = 1;
}
```

$\Longrightarrow$

```
onGet(a) {
    X[h(a)] = 1;
}
```


X :
1
2
N
stream
a
b
c
$\{h(a), h(b), h(c)\}$ – indep

$$P[X[i]=0] = \left(\frac{N-1}{N}\right)^m = \left(1-\frac{1}{N}\right)^m$$

$$L = \sum_{i=1}^{N} [\![X[i]=0]\!] \quad (= \text{number of } i \text{ s.t. } X[i]=0)$$

0 ↙ ↘ 1

$$\underline{E[L]} = \sum_{i=1}^{N} E([\![X[i]=0]\!]) = \sum_{i=1}^{N} Pr[X[i]=0] = \underline{N\left(1-\frac{1}{N}\right)^m}$$

$$E[L] = N\left(1 - \frac{1}{N}\right)^{m}$$

m - number of distinct elements

Method of first moment

u = number of empty urns.

idea: $u \simeq E[L]$

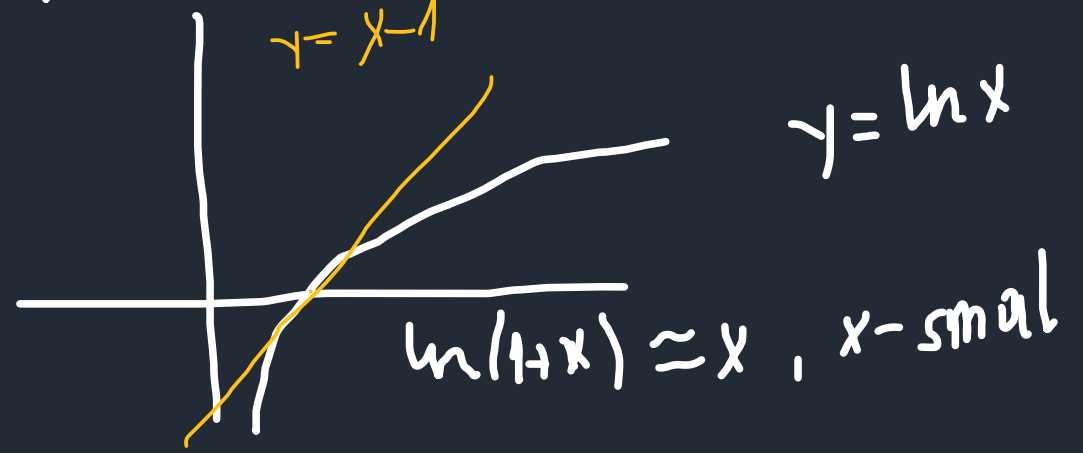
1 0 1 0 1 1 0 0 0 1

$$u = N\left(1 - \frac{1}{N}\right)^{m}$$

$$\frac{u}{N} = \left(1 - \frac{1}{N}\right)^{m}$$

$$\ln\left(\frac{u}{N}\right) = m \ln\left(1 - \frac{1}{N}\right)$$

$$\hat{m} = \frac{\ln\left(\frac{u}{N}\right)}{\ln\left(1 - \frac{1}{N}\right)}$$



$$\ln\left(1-\frac{1}{N}\right) \simeq -\frac{1}{N}$$

$$\hat{m} = \frac{\ln\left(\frac{u}{N}\right)}{-\frac{1}{N}} = -N\ln\left(\frac{u}{N}\right) = N\ln\left(\frac{N}{u}\right)$$

estimator of m

$$\hat{m} = N\ln\left(\frac{N}{u}\right)$$!!!

$u_1, u_2, u_2 \ldots$

m balls

Problem: $u = 0$ $\quad \hat{m} = \infty$

Coupon Collector Problem

$m \simeq n\ln n$

$m \ll n\ln n$

$m < N \rightarrow$ no problem

if $m = N$: $\quad E[L] = N\left(1 - \frac{1}{N}\right)^N \simeq$

$\simeq N \cdot e^{-1} \simeq \frac{N}{3}$

Example: $N = 100$

$\ln N \simeq 4.6$

$N \cdot \ln N \simeq 460$

$m \leq 250$; $u \leq 300$

Precision $\hat{m}$?

EXERCISES:

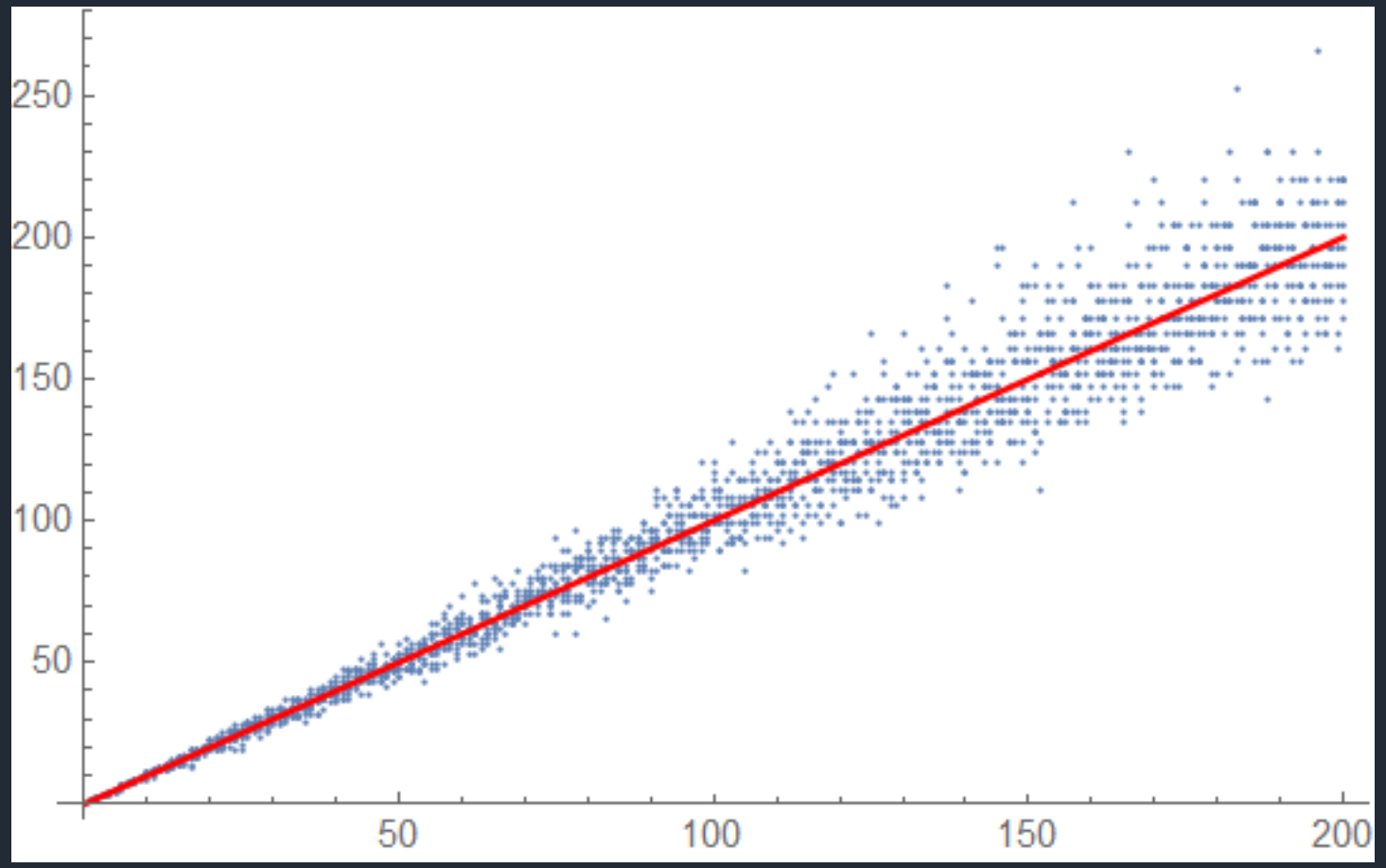
① implement LC.

② estimate

$\mathrm{var}(L) = E[L^2] - E[L]^2$

③ $\frac{\mathrm{var}(L)}{E[L]^2}$ Chebyshev ineq.

Precision of L.C for $N=100$ and $m = 1, 2, \ldots, 200$



(10 simulations for each k)

LC : good $m \leq 10^{10}$

$16^{10} = 10 \cdot \underbrace{10^9}_{1.6B}$ we need $\approx$ 1 GM Memory for X.

{ collisions in Geneva : $10^{15} \approx m$ to beg for LC.

PROBABILISTIC COUNTERS

- classical counter
  - $n = 0$;
  - onGet(x) $\{ n++ \}$

$n \leq N$ $\qquad n = (a_k a_{k-1} \dots a_2 a_1 a_0)_2 =$ $\qquad a_i \in \{0,1\}$

$$= a_0 + a_1 2 + a_2 \cdot 2^2 + \dots + a_k \cdot 2^k$$

$$\leq 1 + 2 + \dots + 2^k \leq 2^{k+1} - 1 \leq N$$

$$2^{k+1} \leq N + 1$$

$$k \leq \log_2(N+1) - 1$$

we need $\pm \log_2(N)$ bits

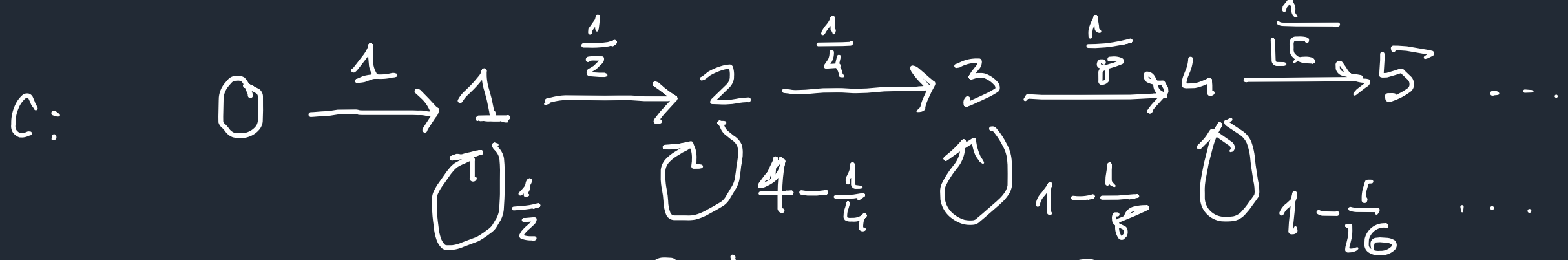
Suppose that we do not need total precision!

$$n = 23645 \approx 23600 \approx 23700$$

$$n \approx 23500$$

MORRIS COUNTER

```
C = 0;
onGet (x) {
    if (random() < (1/2)^C) {
        C++
    }
}
```



if (count) $< \left(\frac{1}{2}\right)^c$) { c++ }

Morris counter

Tw: $\boxed{E[2^{c_n}] = n+1}$

c_n = the value of c after n increm.

we do it n times

$2^{c_n} \simeq n$

$c_n \simeq \log_2 n$

How many bits we need

$\simeq \log_2(\log_2 n)$ ∎