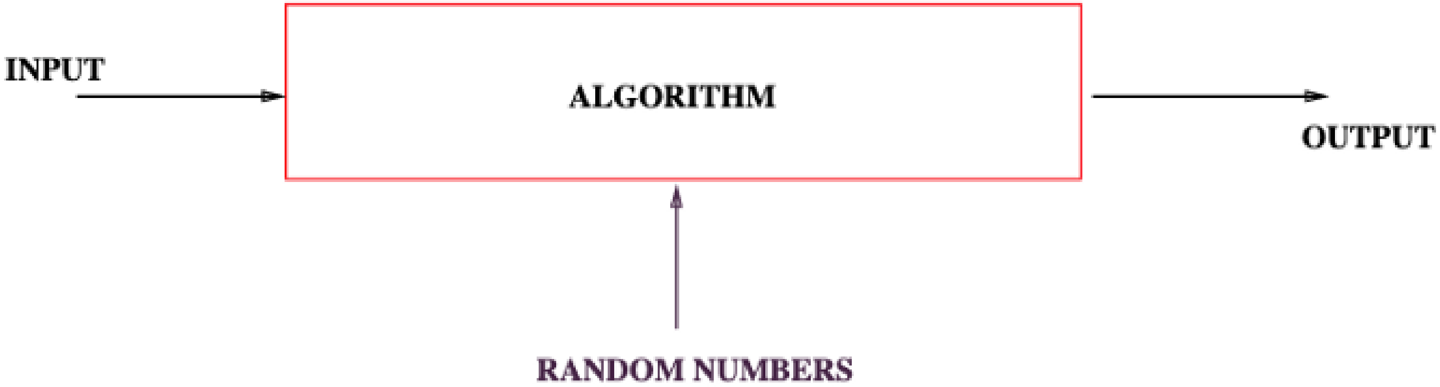
The Birthday Paradox and Coupon Collector Problem

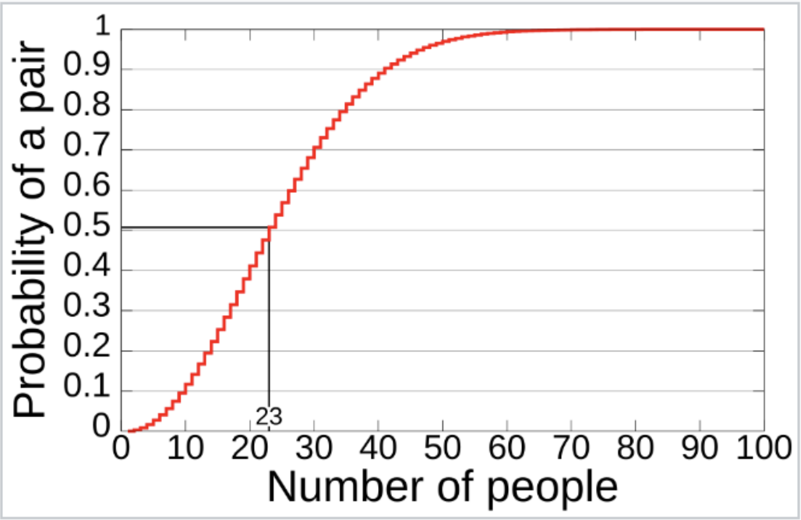
Michael T. Goodrich
CS 263
Univ. of California, Irvine



Some slides adapted from Rutgers University

The Birthday Paradox

- How many must be in a room before there is a 50% probability that two share the same birthday?
  - Answer: 23 individuals are required to reach a 50% probability of a shared birthday.
  - This seems wrong at first glance, but it is, in fact, true.



The computed probability of at least two people sharing the same birthday versus the number of people

The Birthday Paradox

More generally, if there are m people and n possible birthdays then the probability that all m have different birthdays is

$$\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot\left(1-\frac{3}{n}\right)\cdots\left(1-\frac{m-1}{n}\right)=\prod_{j=1}^{m-1}\left(1-\frac{j}{n}\right).$$

Using that $1-k/n \approx \mathrm{e}^{-k/n}$ when k is small compared to n, we see that if m is small compared to n then

$$\begin{aligned}\prod_{j=1}^{m-1}\left(1-\frac{j}{n}\right) &\approx \prod_{j=1}^{m-1}\mathrm{e}^{-j/n}\\ &= \exp\left\{-\sum_{j=1}^{m-1}\frac{j}{n}\right\}\\ &= \mathrm{e}^{-m(m-1)/2n}\\ &\approx \mathrm{e}^{-m^2/2n}.\end{aligned}$$

Application: Hash Collisions

- A cryptographic hash collision is when two different inputs produce the exact same fixed-length hash output, creating a major security weakness because attackers can forge data or signatures by substituting malicious data for legitimate data with the same hash.

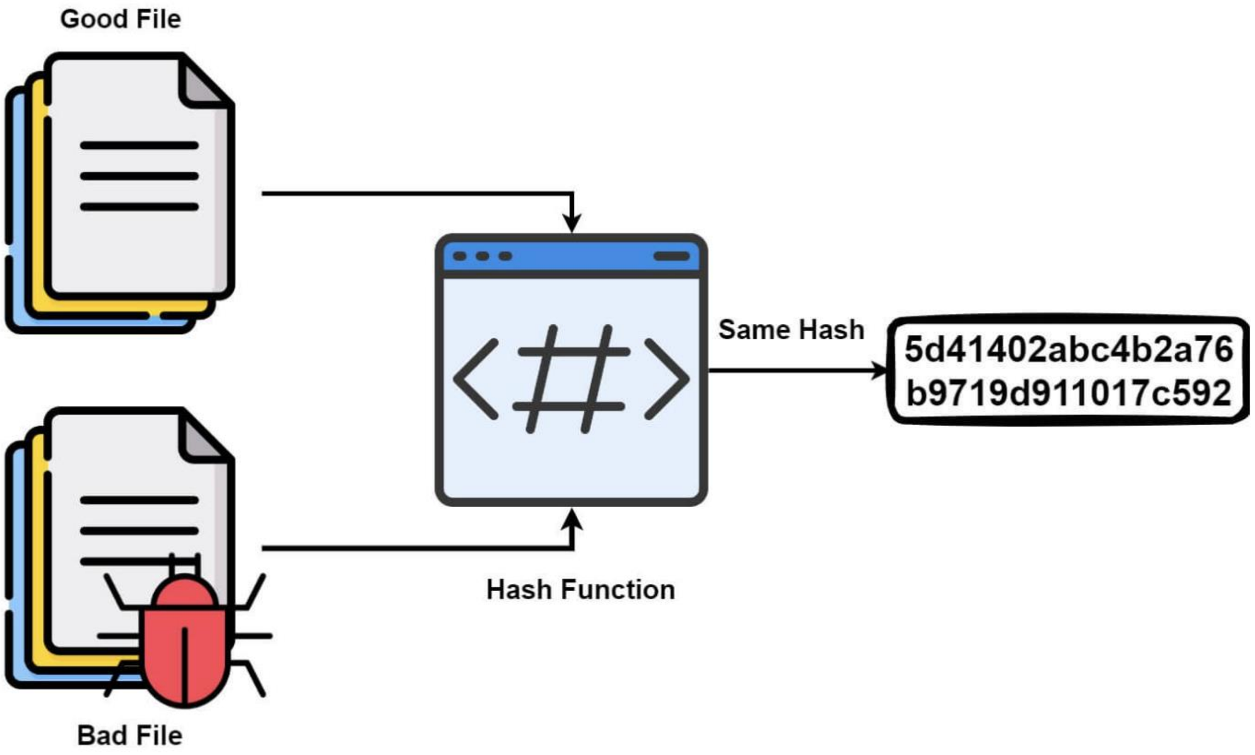


Image from https://library.mosse-institute.com/articles/2023/08/collision-birthday-attack.html

Birthday Attack

- As shown above, the probability is roughly

$$e^{-m^2/2n}$$

- In terms of bits, if the hash function uses k bits, then to get 50% probability you just need

$$m \sim= (2^k)^{1/2} = 2^{k/2} \text{ tries.}$$

- For example, if k=128, as in MD5, you only need roughly 2^{64} tries to find a collision.
- This is not considered secure by modern standards.

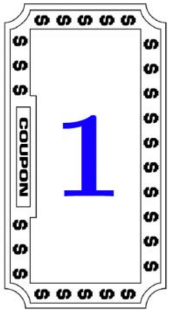
Coupon Collector Problem

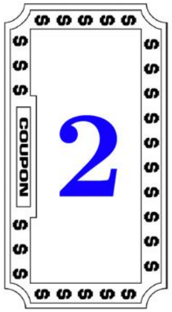
There are *n* different types of coupons.

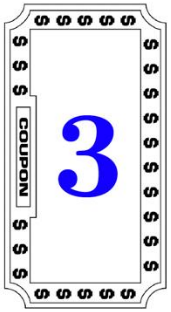
Each purchase comes with one random coupon.

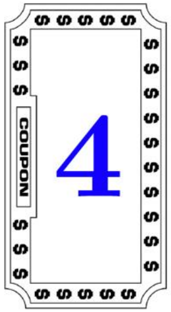
Every coupon is equally likely to appear.

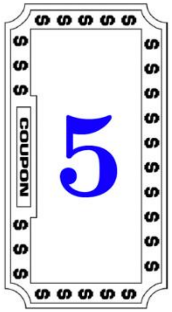
How many purchases needed to collect all coupons?











No definite answer

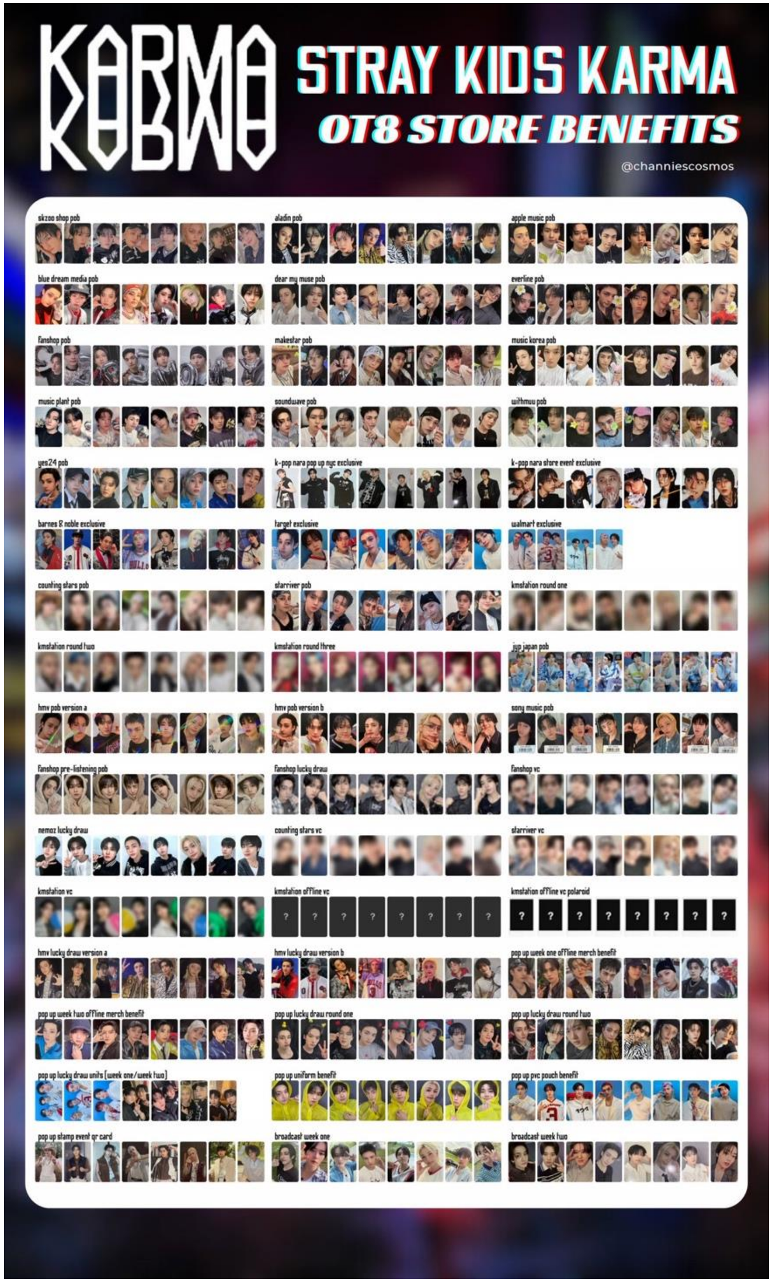
The answer is random:

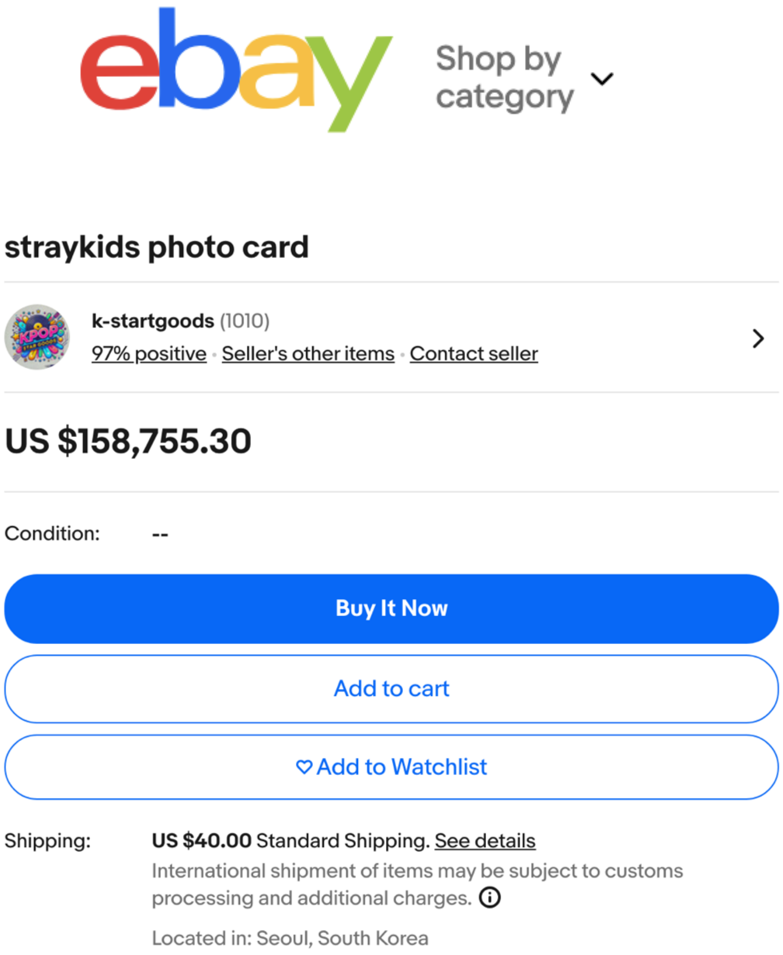
- If lucky, then n purchases are enough.
- If unlucky, then no purchases will be enough.

More suitable question:

How many purchases needed on average to collect all coupons?

Example – K-Pop Photocards





straykids photo card

k-startgoods (1010)

97% positive · Seller's other items · Contact seller

US $158,755.30

Condition: --

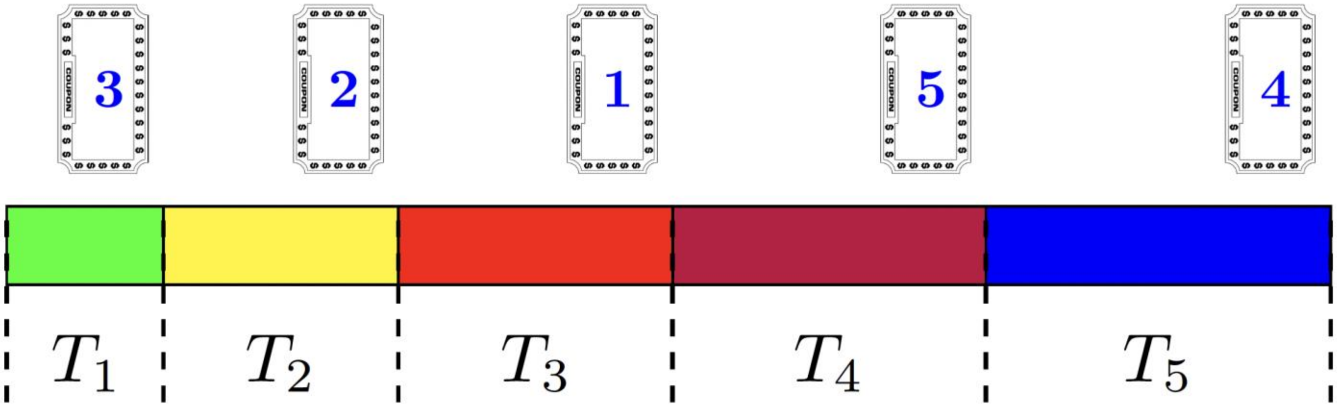
Shipping: **US $40.00** Standard Shipping. See details

International shipment of items may be subject to customs processing and additional charges.

Located in: Seoul, South Korea

Change of perspective

- $T_1 \;=\;$ purchases needed to get 1 coupon.
- $T_2 \;=\;$ purchases needed to get 2 coupons after collecting 1 coupon.
- $T_k \;=\;$ purchases needed to get k coupons after collecting $k-1$ coupons.



Linearity of expectation

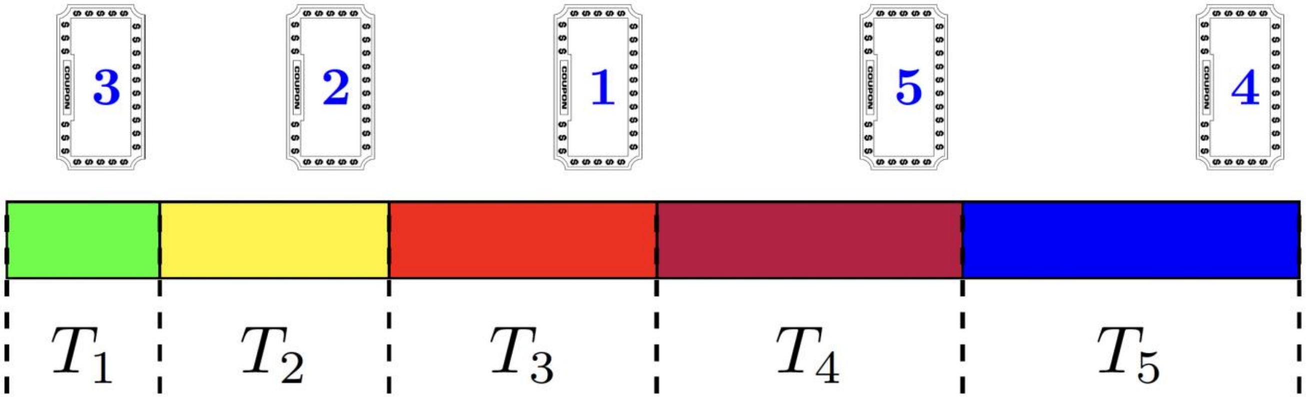
Then to collect all n coupons, we need

$$T = T_1 + T_2 + \ldots + T_n.$$

So

$$E[T] = E[T_1] + E[T_2] + \ldots + E[T_n].$$

This is called linearity of expectations.



Geometric random variable comes back

After collecting $k - 1$ coupons, for next purchase:

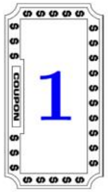
- Failure = getting coupon you already have.

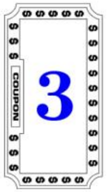
  There are $k - 1$ choices.

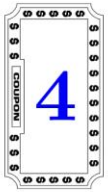
- Success = getting a new coupon.

  There are $n - k + 1$ choices.

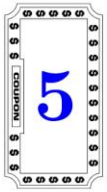
Have:







Don't Have:



Geometric random variable comes back

Then T_k is geometric random variable with

success probability $p = \dfrac{n-k+1}{n}$.

So $$E[T_k] = \frac{1}{p} = \frac{n}{n-k+1}.$$

Solution to coupon's collector problem

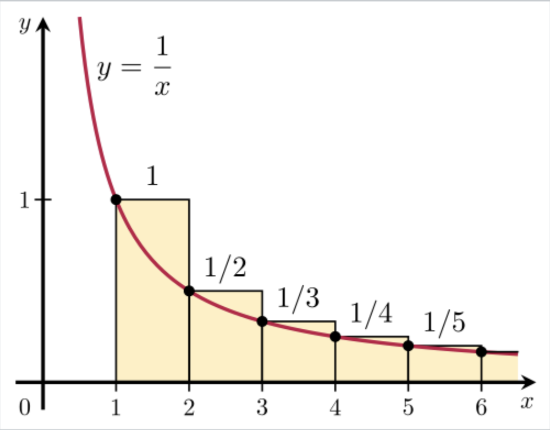
$$\begin{aligned} E[T] &= E[T_1] + E[T_2] + E[T_3] + \ldots + E[T_n] \\ &= \frac{n}{n} + \frac{n}{n-1} + \frac{n}{n-2} + \ldots + \frac{n}{1} \\ &= n\left(\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \ldots + \frac{1}{1}\right) \\ &= n\left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}\right). \end{aligned}$$

This is harmonic series, and

$$E[T] \quad \text{is approximately} \quad n \int_1^n \frac{1}{x}\,dx \quad = \quad n \log n.$$

Precise Expected Time for Coupon Collecting

$$\begin{aligned}
\mathrm{E}(T) &= \mathrm{E}(t_1 + t_2 + \cdots + t_n) \\
&= \mathrm{E}(t_1) + \mathrm{E}(t_2) + \cdots + \mathrm{E}(t_n) \\
&= \frac{1}{p_1} + \frac{1}{p_2} + \cdots + \frac{1}{p_n} \\
&= \frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1} \\
&= n \cdot \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \right) \\
&= n \cdot H_n.
\end{aligned}$$



Graph demonstrating a connection between harmonic numbers and the natural logarithm. The harmonic number H_n can be interpreted as a Riemann sum of the integral:

$$\int_1^{n+1} \frac{dx}{x} = \ln(n+1).$$

Here H_n is the n-th harmonic number. Using the asymptotics of the harmonic numbers, we obtain:

$$\mathrm{E}(T) = n \cdot H_n = n \ \ln \ n + \gamma n + \frac{1}{2} + O(1/n),$$

where $\gamma \approx 0.5772156649$ is the Euler–Mascheroni constant.

High Probability Bound

- To prove that the **coupon collector** problem requires O(n log n) time with high probability, to collect all n coupons we will actually calculate the probability that a specific number of trials is insufficient to collect all coupons.
- The total time, T, can be viewed as a sum of n geometric random variables (which are not identically distributed), so we will apply a standard Chernoff bound here using an *independent count technique*, where we fix the number of trials and analyze the number of times *each specific coupon* appears.

Independent Count Technique

- Let n be the number of distinct coupons.
- We run the coupon collection for t trials, where:

$$t = c * n \ln n,$$

where c is a constant we will choose later.

- We want to prove that after t trials, the probability that we have **not** collected all n coupons is very small (i.e., bounded by 1/n).

Defining the Random Variables

- Let X_i be a random variable representing the number of times the **i-th coupon** is collected during these t trials. This is a sum of t **independent** 0-1 random variables, each of which is 1 with probability p = 1/n.
- Then

$$\mu = E[X_i] = t \cdot p = (cn \ln n) \cdot \frac{1}{n} = c \ln n$$

- We will actually compute the probability that we collect fewer than (c/2)ln n copies of the i-th coupon.

Applying a Chernoff Bound

- Use a simplified Chernoff bound, with δ=1/2.

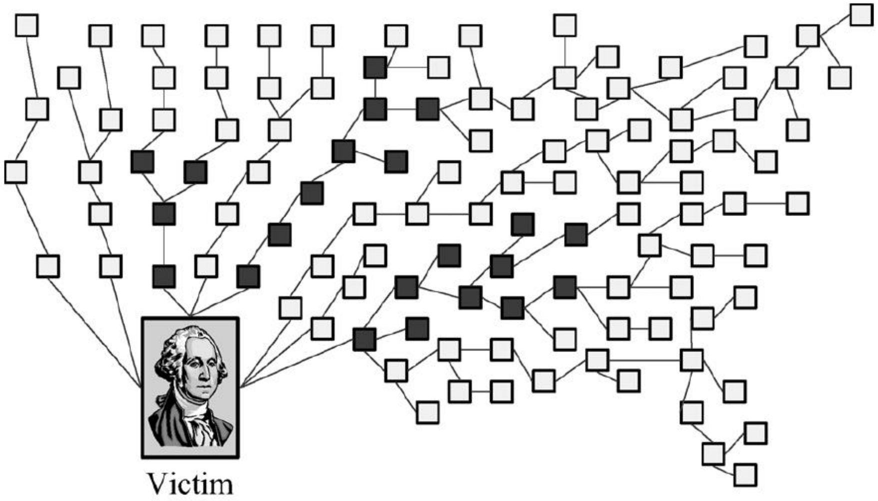
$$P(X_i \leq (1-\delta)\mu) \leq e^{-\frac{\delta^2 \mu}{2}}$$

- So, with μ=c*ln n, this probability is at most

$$e^{-(c/8)\ln n} = n^{-c/8}$$

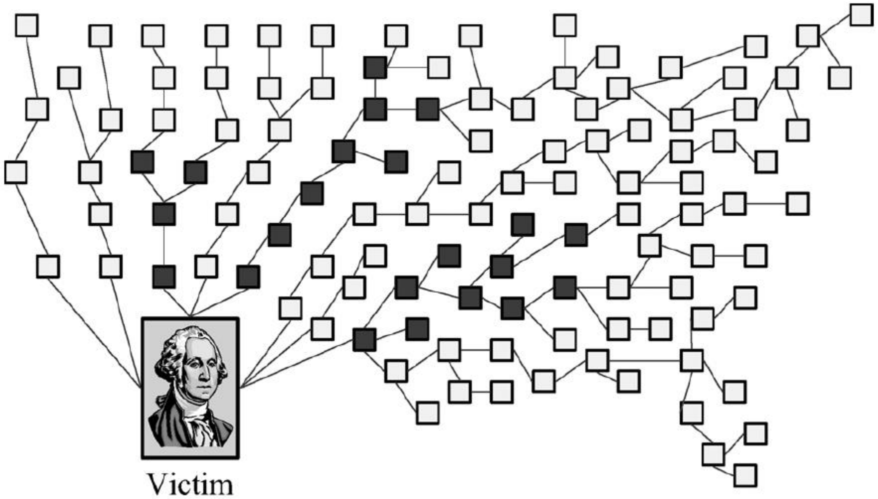
- Taking c=16, we get a failure probability of $1/n^2$.
- Thus, by a union bound for all n coupons, we will collect at least (c/2)ln n copies of each coupon with probability at least 1-1/n.

Application – IP Traceback



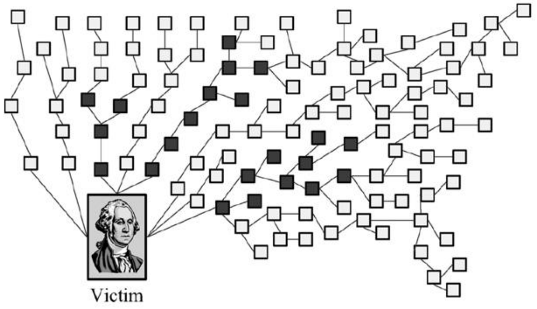
- During a distributed denial-of-service (DDOS) attack, zombie hosts send a lot of message requests to a victim.
- The goal in the traceback problem is to identify the leaves of the attack tree, that is, the routers upstream from the victim closest to attack by having routers mark packets with their ID with some probability p (e.g., p=1/20).

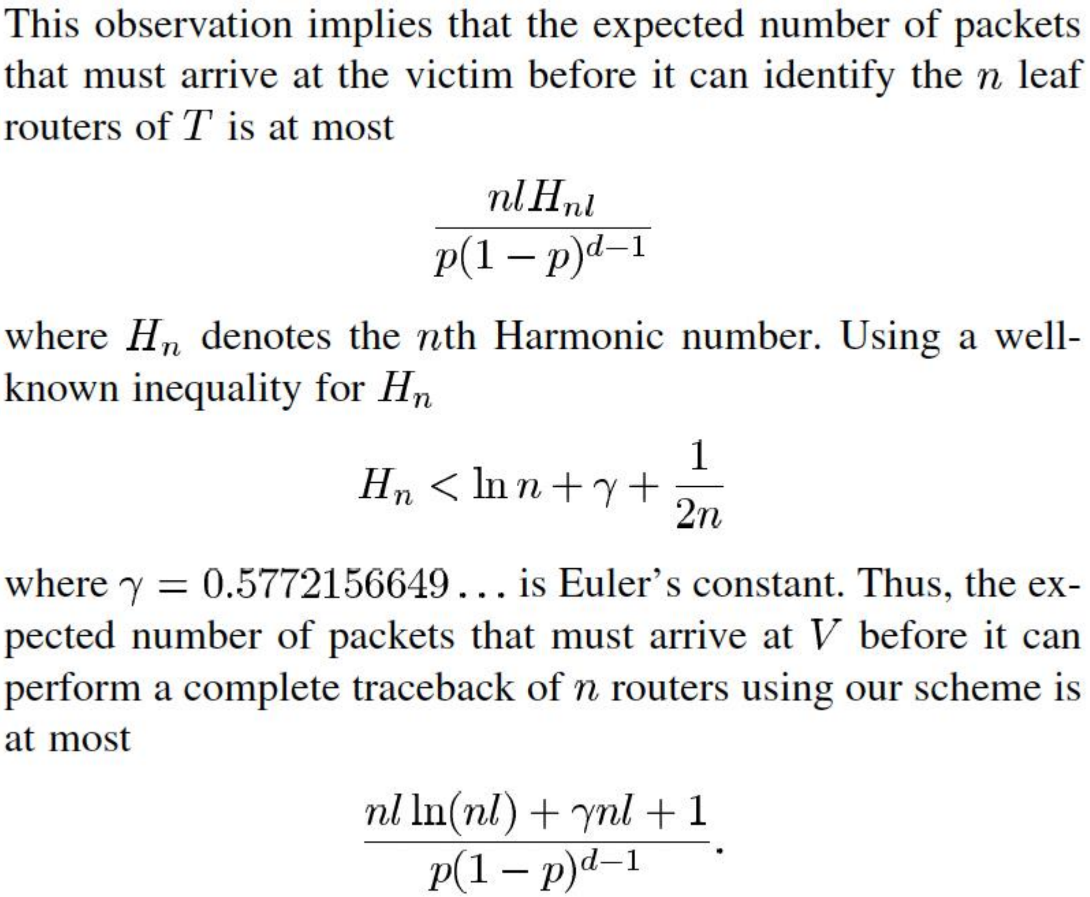
Application – IP Traceback



- To save bits, each router breaks its ID into *l* pieces and randomly chooses which piece to use to mark a packet.
- In order to traceback the attack nodes, the victim needs to collect at least one packet marked by each intermediate router.
- The probability that a packet at distance d reaches the victim is $p(1-p)^{d-1}$.

IP Traceback is Coupon Collecting





This observation implies that the expected number of packets that must arrive at the victim before it can identify the n leaf routers of T is at most

$$\frac{nlH_{nl}}{p(1-p)^{d-1}}$$

where H_n denotes the nth Harmonic number. Using a well-known inequality for H_n

$$H_n < \ln n + \gamma + \frac{1}{2n}$$

where $\gamma = 0.5772156649\ldots$ is Euler's constant. Thus, the expected number of packets that must arrive at V before it can perform a complete traceback of n routers using our scheme is at most

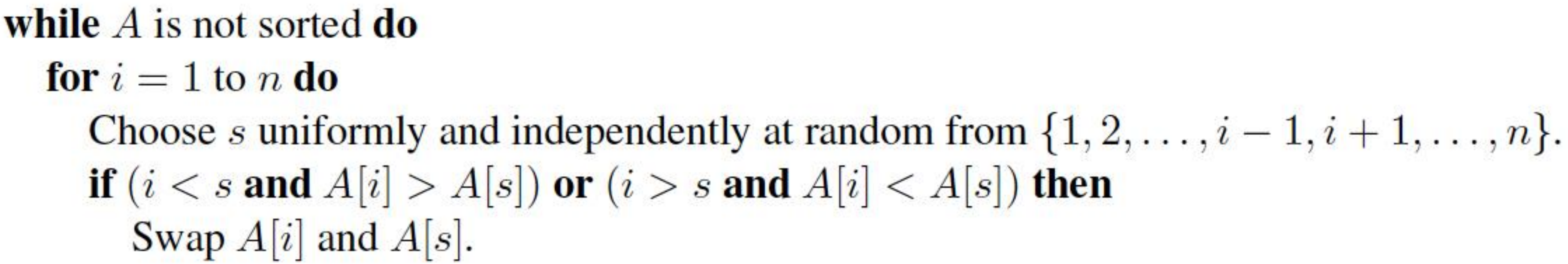
$$\frac{nl\ln(nl) + \gamma nl + 1}{p(1-p)^{d-1}}.$$

The Attacker’s Problem

- From the attacker’s perspective, finding all the zombie computers, which are vulnerable to a given intrusion attack is also a coupon collection problem.
- Note that in the coupon collection problem the goal is to collect **all** the coupons.
- If we change the goal to that of collecting, e.g., 50% or 75% of the coupons, then this can be done in expected linear time.
- Why?

A Silly (But Instructive) Application

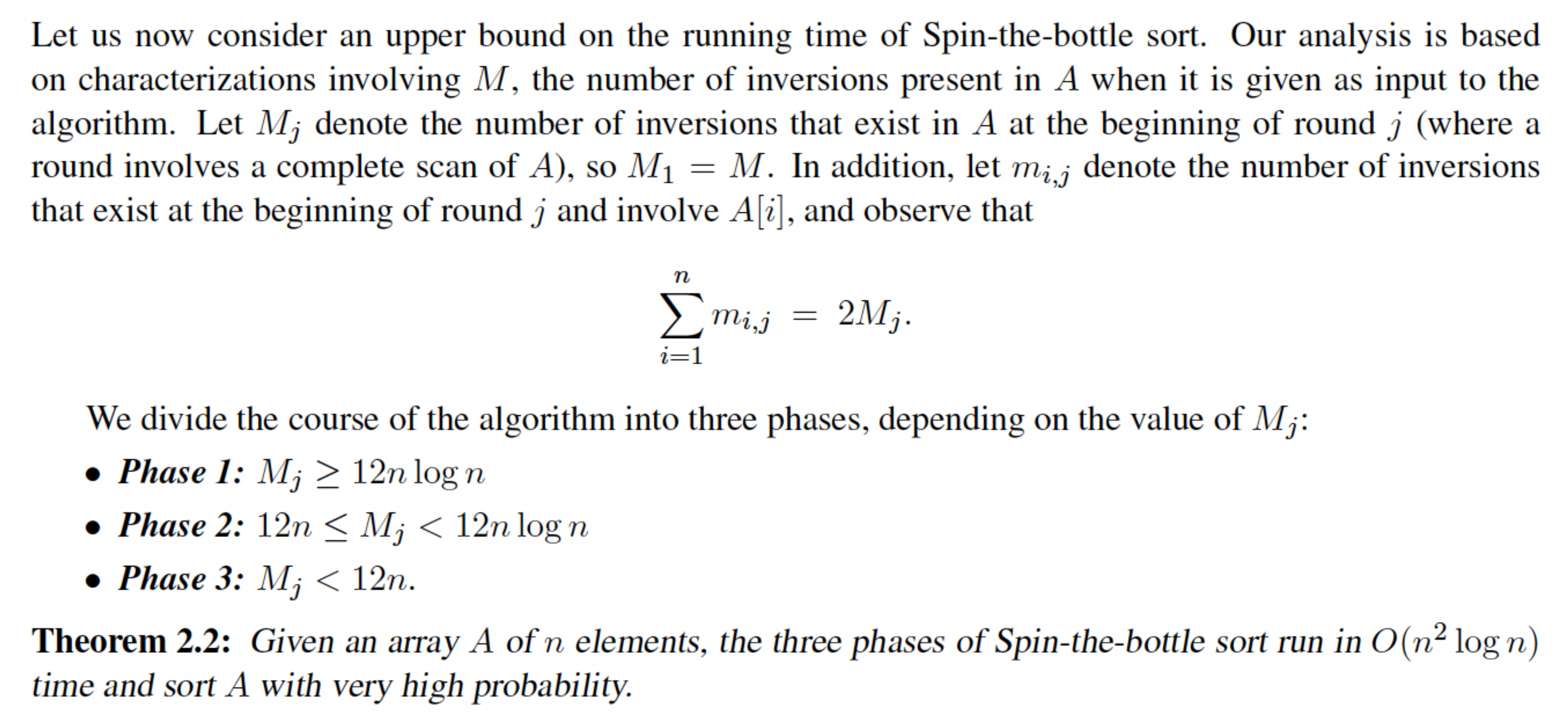
- Spin-the-Bottle Sort:



```
while $A$ is not sorted do
  for $i = 1$ to $n$ do
    Choose $s$ uniformly and independently at random from $\{1, 2, \ldots, i-1, i+1, \ldots, n\}$.
    if $(i < s$ and $A[i] > A[s])$ or $(i > s$ and $A[i] < A[s])$ then
      Swap $A[i]$ and $A[s]$.
```

- What is a high-probability bound for its running time?

Running Time Analysis



Let us now consider an upper bound on the running time of Spin-the-bottle sort. Our analysis is based on characterizations involving M, the number of inversions present in A when it is given as input to the algorithm. Let M_j denote the number of inversions that exist in A at the beginning of round j (where a round involves a complete scan of A), so $M_1 = M$. In addition, let $m_{i,j}$ denote the number of inversions that exist at the beginning of round j and involve $A[i]$, and observe that

$$\sum_{i=1}^{n} m_{i,j} \; = \; 2M_j.$$

We divide the course of the algorithm into three phases, depending on the value of M_j:

- ***Phase 1:*** $M_j \geq 12n \log n$
- ***Phase 2:*** $12n \leq M_j < 12n \log n$
- ***Phase 3:*** $M_j < 12n$.

Theorem 2.2: *Given an array A of n elements, the three phases of Spin-the-bottle sort run in $O(n^2 \log n)$ time and sort A with very high probability.*

Phase 1. Let X_j be a random variable that equals the number of inversions resolved in round j of Phase 1, and let $X_{i,j}$ denote an indicator random variable that is 1 iff we perform a comparison in iteration (round) j of the algorithm between $A[i]$ and an element that caused an inversion with $A[i]$ at the beginning of round j. Thus,

$$X_j \geq \frac{\sum_{i=1}^{n} X_{i,j}}{2},$$

since each inversion involves two elements of A. Each of the $X_{i,j}$'s are independent. Furthermore,

$$E(X_{i,j}) = \frac{m_{i,j}}{n-1},$$

where $m_{i,j}$ denotes the number of inversions that exist at the beginning of round j and involve $A[i]$. Therefore,

$$E(X_j) \geq (1/2) \sum_{i=1}^{n} \frac{m_{i,j}}{n-1} = M_j/(n-1),$$

where M_j is the number of inversions in A that exist at the beginning of round j. Thus, by a well-known Chernoff bound,

$$\begin{aligned} \Pr(X_j < M_j/2(n-1)) &\leq \left(\frac{e^{-1/2}}{(1/2)^{1/2}}\right)^{M_j/(n-1)} \\ &\leq 2^{-M_j/3(n-1)} \\ &\leq n^{-4}, \end{aligned}$$

since we are in Phase 1. So we may assume with probability at least $1 - c/n^3$ that the following recurrence relation holds during Phase 1, for all $1 \leq j \leq cn$, for any constant $c \geq 1$:

$$M_{j+1} \leq M_j - \frac{M_j}{2n}.$$

Therefore, with probability at least $1 - 4/n^3$, there are at most $4n$ rounds during Phase 1 of Spin-the-bottle sort, since $M_1 = M < n^2$ and $M_j \geq 12n \log n$, for all j during Phase 1. That is, with very high probability, Phase 1 runs in $O(n^2)$ time.

Phase 2. For this phase, let X_j and $X_{i,j}$ denote random variables defined as in our analysis of Phase 1, with the index j reset to 1 for Phase 2. In this case,

$$E(X_j) \geq M_j/(n-1) \geq 12.$$

Thus, by a similar Chernoff bound used for analyzing Phase 1,

$$\begin{aligned} \Pr(X_j < 6) &\leq \Pr(X_j < M_j/2(n-1)) \\ &\leq 2^{-M_j/3(n-1)} \\ &\leq 2^{-4}, \end{aligned}$$

since we are in Phase 2. That is, with probability $1/16$ we resolve fewer than 6 inversions in round j of Phase 2. Call round j a ***failure*** in this case, and call it a ***success*** if it resolves at least 6 inversions. Let Y_j be an indicator random variable that is 1 iff we resolve fewer than 6 inversions in round j of Phase 2, or, if j is larger than the number of rounds in Phase 2, then let Y_j be an independent random variable that is 1 with probability $1/16$. Thus, the number of failure rounds in the first at most $4n \log n$ rounds of Phase 2 is at most

$$Y = \sum_{j=1}^{4n \log n} Y_j.$$

Note that $E(Y) = (1/4)n \log n$. Thus, by a standard Chernoff bound,

$$\begin{aligned} \Pr(Y > 2n \log n) &= \Pr(Y > 8(1/4)n \log n) \\ &\leq \left(\frac{e^7}{8^8}\right)^{(1/4)n \log n} \\ &\leq 2^{-2n \log n} \\ &= n^{-2n}. \end{aligned}$$

Note, in addition, that there can be, in total, at most $2n \log n$ successful rounds in Phase 2. Thus, with very high probability, there are only $O(n \log n)$ rounds in Phase 2. That is, with very high probability, Phase 2 runs in $O(n^2 \log n)$ time.

Phase 3. The analysis for this phase is similar to that for the coupon collector's problem (e.g., see [7]). At the start of this phase, there are fewer than $12n$ inversions that remain in A. Note that, for any such inversion, χ, the probability that χ is resolved in a round of Phase 3 is at least[6] $1/n$. Let Z_χ^r be the event that χ is not resolved after r rounds of Phase 3. Thus,

$$\Pr(Z_\chi^r) \leq \left(1 - \frac{1}{n}\right)^r \leq e^{-r/n}.$$

Let R denote the number of rounds needed to resolve all the inversions in Phase 3. Then, for $c \geq 2$,

$$\begin{aligned} \Pr(R > cn \ln n) &\leq \Pr\left(\bigcup_\chi Z_\chi^{cn \log n}\right) \\ &\leq \sum_\chi \Pr\left(Z_\chi^{cn \log n}\right) \\ &\leq \frac{12}{n^{c-1}}. \end{aligned}$$

Thus, with very high probability, R is $O(n \log n)$; hence, with very high probability, Phase 3 runs in $O(n^2 \log n)$ time. This completes the proof.