Randomness and Computation  
or, “Randomized Algorithms”

Heng Guo  
(Based on slides by M. Cryan)

---

**warm-up: Birthday Paradox**

30 people in a room. What is the probability they share a birthday?

- Assume everyone is equally likely to be born any day (*uniform at random*). Exclude Feb 29 for neatness.
- Also assume that the birthdays are mutually independent. (E.g. no twins)

Probability $p_{30\text{diff}}$ that all birthdays are different can be directly calculated

$$p_{30\text{diff}} = \frac{30! \left(\frac{365}{30}\right)^{30}}{365^{30}}.$$ 

Alternatively, we can also use the principle of deferred decision. “Generate” the birthdays one by one

$$p_{30\text{diff}} = \prod_{i=1}^{30} \frac{365 - (i - 1)}{365} = \prod_{i=1}^{30} \left(1 - \frac{(i - 1)}{365}\right) = \prod_{j=1}^{29} \left(1 - \frac{j}{365}\right).$$

Recall that $1 + x < e^x$ for all $x \in \mathbb{R}$. Hence $1 - \frac{j}{365} < e^{-j/365}$ for any $j$.

$$p_{30\text{diff}} < \prod_{j=1}^{29} e^{-j/365} = \left(\prod_{j=1}^{29} e^{-j}\right)^{\frac{1}{29}} = \left(e^{-\frac{29}{2}}\right)^{\frac{1}{29}} = \left(e^{-435}\right)^{\frac{1}{29}},$$

where the last step used $\sum_{j=1}^{n} j = \frac{n(n+1)}{2}$. 

---
warm-up: Birthday Paradox

So far we have

\[ p_{30 \text{diff}} < (e^{-435})^{\frac{30}{365}} < e^{-1.19} \approx 0.3042. \]

This approximation is pretty close, as \( p_{30 \text{diff}} \approx 0.2937 \).

Continuing,

With probability of at least 0.7, two people at the party share a birthday.

More general framework: \( n \) birthday options, \( m \) persons

Probability \( p_{\text{all-m-diff}} \) that all are different is

\[ p_{\text{all-m-diff}} = \prod_{j=1}^{m} \left(1 - \frac{j-1}{n}\right) = \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right). \]

RC (2019/20) – Lecture 9 – slide 4

warm-up: General Birthday Paradox

\( n \) birthday options, \( m \) persons

Probability \( p_{\text{all-m-diff}} \) that all are different is

\[ p_{\text{all-m-diff}} = \prod_{j=1}^{m} \left(1 - \frac{j-1}{n}\right) = \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right). \]

Continuing,

\[ p_{\text{all-m-diff}} \leq \prod_{j=1}^{m-1} e^{-j/n} = \left( \prod_{j=1}^{m-1} e^{-j} \right)^{\frac{1}{2}} = \left( e^{-\sum_{j=1}^{m-1} j} \right)^{\frac{1}{2}} = e^{-\frac{(m-1)m}{2n}}, \]

approximately \( e^{-m^2/2n} \).

RC (2019/20) – Lecture 9 – slide 5
warm-up: General Birthday Paradox

$n$ birthday options, $m$ persons

Probability $p_{\text{all}-m-\text{diff}}$ that all are different is

$$p_{\text{all}-m-\text{diff}} = \prod_{j=1}^{m-1} \left(1 - \frac{(j-1)}{n}\right) = \prod_{j=1}^{m-1} \left(1 - \frac{j}{n}\right).$$

Continuing,

$$p_{\text{all}-m-\text{diff}} \leq \prod_{j=1}^{m-1} e^{-j/n} = \left(\prod_{j=1}^{m-1} e^{-j}\right)^{1/n} = \left(e^{-\sum_{j=1}^{m-1} j}\right)^{1/n} = e^{-\frac{(m-1)m}{2n}},$$

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

Suppose we set $m = \lfloor \sqrt{n} \rfloor$, then $e^{-m^2/2n}$ becomes $e^{-0.5 \sim 0.6}$.

The paradox

$n$ birthday options, $m$ persons

Deterministically, there is guaranteed to have a collision (two persons sharing the same birthday) if and only if $m \geq n + 1$.

Randomly, with $m = \Omega(\sqrt{n})$, the probability of a collision is very high.

For example, if $n = 365$ and $m = 57$, $p_{\text{diff}} < 1\%$.

Balls into Bins

- $m$ balls, $n$ bins, and balls thrown uniformly at random and independently into bins (usually one at a time).
- Magic bins with no upper limit on capacity.
- Can be viewed as a random function $[m] \to [n]$.
- Common model of random allocations and their effects on overall load and load balance, typical distribution in the system.

Many related questions:
- How many balls do we need to cover all bins? (Coupon collector, surjective mapping)
- How many balls will lead to a collision? (Birthday paradox, injective mapping)
- What is the maximum load of each bin? (Load balancing)
Load balancing

Load balancing is a very important problem, especially for networks. “Balls into Bins” is a simplified model for hashing.

A success story worth mentioning: Akamai

Consistent Hashing and Random Trees, STOC 1997
Karger, Lehman, Leighton, Levine, Lewin, Panigrahy

One year later, Leighton and Lewin co-founded Akamai based on this technique. They created the “Content Delivery Network” (CDN) industry. Many well-known services, including Apple, Facebook, Google / Youtube, Steam, Netflix, (partly) rely on it.

Balls into Bins maximum load

We aim to bound the maximum load of the “Balls into Bins” model in the case of \( m = n \). For any bin \( i \in [n] \), its load, denoted \( X_i \), has expectation

\[
E[X_i] = \sum_{j=1}^{n} E[X_{ij}] = 1.
\]

Let \( X_i > T \) be our “bad events” for some threshold \( T \). Then to get a whp result via union bound, we need to at least upper bound the bad event like

\[
\Pr[X_i > T] \leq \frac{1}{n^2}.
\]

Thus Markov inequality is not good enough, nor is Chebyshev (\( \text{Var}[X_i] = \sum_{j=1}^{n} \text{Var}[X_{ij}] = 1 - \frac{1}{n} \)).

Chernoff bounds actually work here, since \( X_i \)'s are negatively correlated. We will do a quicker “ad hoc” analysis for the upper bound first.
Lemma (5.1)
Let $n$ balls be thrown independently and uniformly at random into $n$ bins. Then for sufficiently large $n$, the maximum load is bounded above by $\frac{3 \ln(n)}{\ln \ln(n)}$ with probability at least $1 - \frac{1}{n}$.

Proof: The probability that bin $i$ receives $\geq M$ balls is at most

$$\binom{n}{M} \frac{n^{n-M}}{n^n} = \binom{n}{M} \frac{1}{n^M}.$$ 

Binomial coefficient satisfies

$$\left( \frac{n}{M} \right)^M \leq \binom{n}{M} \leq \frac{n^M}{M!} \leq \left( \frac{e n}{M} \right)^M.$$ 

Bin $i$ gets $\geq M$ bins with probability at most $\left( \frac{e n}{M} \right)^M = \left( e \frac{n}{M} \right)^M$. 

Proof of Lemma 5.1 cont’d.
Bin $i$ gets $\geq M$ bins with probability at most $\left( \frac{e n}{M} \right)^M$. 

RC (2019/20) – Lecture 9 – slide 10
Balls into Bins maximum load

Proof of Lemma 5.1 cont’d.

Bin \( i \) gets \( \geq M \) bins with probability at most \( \left( \frac{e}{M} \right)^M \).

Set \( M := \frac{3 \ln(n)}{\ln(\ln(n))} \). Then the probability that any bin gets \( \geq M \) balls is (using the Union bound) at most

\[
 n \cdot \left( \frac{e \cdot \ln(n)}{3 \ln(n)} \right)^{\frac{3 \ln(n)}{\ln(\ln(n))}} \leq n \cdot \left( \frac{\ln(n)}{\ln(n)} \right)^{\frac{3 \ln(n)}{\ln(\ln(n))}} = e^{\ln(n)} \left( \frac{\ln(n)}{\ln(n)} \right)^{\frac{3 \ln(n)}{\ln(\ln(n))}}.
\]

Again using properties of \( \ln \), this expands as

\[
e^{\ln(n)} \left( e^{\ln(\ln(n)) - \ln(n)} \right)^{\frac{3 \ln(n)}{\ln(\ln(n))}} = e^{\ln(n)} \left( e^{-3 \ln(n) + 3 \ln(\ln(n))} \right).
\]

Balls into Bins maximum load

Proof of Lemma 5.1 cont’d.

Grouping the \( \ln(n) \)'s in the exponents, and evaluating, we have

\[
e^{-2 \ln(n)} \cdot e^{\frac{3 \ln(n) \ln(\ln(n))}{\ln(\ln(n))}} = n^{-2} \cdot n^{\ln(\ln(n))}.
\]

If we take \( n \) "sufficiently large" (\( n \geq e^4 \) will do it), then \( \frac{\ln(\ln(n))}{\ln(n)} \leq 1/3 \), hence the probability of some bin having \( \geq M \) balls is at most

\[
n^{-1}.
\]
Balls into Bins maximum load

Proof of Lemma 5.1 cont’d.
Grouping the ln(n)s in the exponents, and evaluating, we have
\[ e^{-2\ln(n)} \cdot e^{\frac{\ln(n) \cdot \ln(\ln(n))}{\ln(n)}} = n^{-2} \cdot n^{\frac{\ln(n) \cdot \ln(\ln(n))}{\ln(n)}}. \]
If we take n “sufficiently large” (n \geq e^{e^d} will do it), then \frac{\ln\ln(n)}{\ln(n)} \leq 1/3, hence the probability of some bin having \geq M balls is at most
\[ n^{-1}. \]
Can derive a matching proof to show that “with high probability” there will be a bin with \Omega(\frac{\ln(n)}{\ln\ln(n)}) balls in it.

The power of two choices

Instead of throwing balls randomly, we throw them sequentially with the following tweak: for each ball, we pick two random choices of bins, and choose the one with the lower load.

Surprisingly, the maximum load in this case is \ln\ln n/\ln 2 \pm O(1) with probability 1 - o(1/n)!
The load reduces from \Theta(\frac{\ln n}{\ln \ln n}) to \Theta(\ln \ln n)!

RC (2019/20) – Lecture 9 – slide 13

The power of two choices

Instead of throwing balls randomly, we throw them sequentially with the following tweak: for each ball, we pick two random choices of bins, and choose the one with the lower load.

Surprisingly, the maximum load in this case is \ln\ln n/\ln 2 \pm O(1) with probability 1 - o(1/n)!
The load reduces from \Theta(\frac{\ln n}{\ln \ln n}) to \Theta(\ln \ln n)!

More generally, we may have \(d \geq 2\) choices, and the resulting maximum load is \ln\ln n/\ln d \pm O(1) with probability 1 - o(1/n).

This is Theorem 17.1 of [MU] (details in Section 17.1/17.2).
Some preliminary observations, definitions

The probability of a specific bin (bin $i$, say) being empty is:

$$
\left(1 - \frac{1}{n}\right)^m \sim e^{-m/n}.
$$

Expected number of empty bins: $\sim ne^{-m/n}$.

Probability $p_r$ of a specific bin having $r$ balls:

$$
p_r = \binom{m}{r} \left(\frac{1}{n}\right)^r \left(1 - \frac{1}{n}\right)^{m-r}.
$$

Note

$$
p_r \sim \frac{e^{-m/n}}{r!} \left(\frac{m}{n}\right)^r.
$$

Definition (5.1)

A discrete Poisson random variable $X$ with parameter $\mu$ is given by the following probability distribution on $j = 0, 1, 2, \ldots$:

$$
\Pr[X = j] = \frac{e^{-\mu} \mu^j}{j!}.
$$
References and Exercises

▶ Sections 5.1, 5.2 of “Probability and Computing” [MU].
▶ On Friday we will do the lower bound and the Poisson approximation. Read Sections 5.3 and 5.4.

Exercises

▶ Exercise 5.3 (balls in bins when $m = c \cdot \sqrt{n}$).
▶ Exercise 5.10 (sequences of empty bins; this is a bit more tricky)