CS70 @ UC Berkeley, Spring 2026

Lecture 19
Some Applications

April 2, 2026

Beating Random Guess

Alice and **Bob** play the following guessing game:

- **Bob** writes down two different numbers in $(0, \ell)$ on two separate cards:

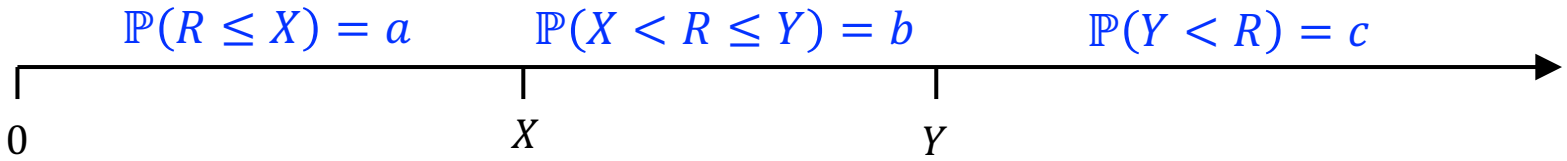
- These numbers are hidden to **Alice**, who picks one of the two cards uniformly at random and looks at the number.
- **Alice** wins if she correctly guesses which of the two cards has a larger number.

Question: Can Alice do better than random guess (i.e., greater than probability ½ of winning?

Beating Random Guess

The answer is Yes! Here is a strategy.

1. Suppose Bob's numbers are $X < Y$.
2. Let A denote the number that Alice picked.
3. Generate a random number $R \in (0, \ell)$.
   - If $R > A$, go with the other card.
   - If $R \leq A$, stick to the original chosen card.



Alice can't compute these probabilities since she doesn't know how Bob generated X and Y. Nevertheless, she can show:

$$\mathbb{P}(\text{Correct}) = \underset{b+c}{\mathbb{P}(\text{Correct} \mid A = X)} \underset{\frac{1}{2}}{\mathbb{P}(A = X)} + \underset{a+b}{\mathbb{P}(\text{Correct} \mid A = Y)} \underset{\frac{1}{2}}{\mathbb{P}(A = Y)}$$

$$= \frac{1}{2}(a + b + b + c) = \frac{1}{2} + \frac{b}{2} > \frac{1}{2} \quad (\text{since } b > 0)$$

Get the Largest Number

Consider a deck of N cards each with a number written on one side, facing down. **Assume:**

- N is large and all numbers are distinct.
- The deck is well shuffled.

Goal: Get the largest number.

Rules:

1. Reveal one card at a time starting from top.
2. STOP at the current card or reveal the next card.
3. If you pass on a card that has been revealed, you can't choose to it later.

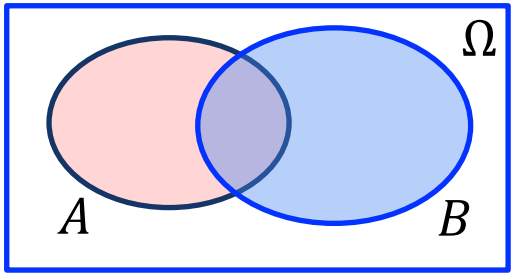
Strategy: Reveal a certain proportion, say p, of the cards and record the largest number, denoted M, you have seen. Then, STOP when you see a number larger than M.

Question: What should p be to maximize your chance of winning?

Solution will be provided in the next lecture.

Union of Events (Last Lecture)

Let A, B be two events on the same probability space.



$$\mathbb{P}(A) = \mathbb{P}(A \backslash (A \cap B)) + \mathbb{P}(A \cap B)$$

$$\mathbb{P}(B) = \mathbb{P}(B \backslash (A \cap B)) + \mathbb{P}(A \cap B)$$

$$\mathbb{P}(A \cup B) = \mathbb{P}(A \backslash (A \cap B)) + \mathbb{P}(B \backslash (A \cap B)) + \mathbb{P}(A \cap B)$$

$$= \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$$

Recall the Inclusion-Exclusion formula from Lecture 15:

Theorem 5 (Inclusion-Exclusion). *Let $A_1, \ldots, A_n$ be arbitrary subsets of the same set Ω. Then,*

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \sum_{S \subseteq \{1,\ldots,n\} : |S| = k} \left| \bigcap_{i \in S} A_i \right|.$$

Replace $|E|$ with $\mathbb{P}(E)$ in both sides to obtain the Inclusion-Exclusion formula for $\mathbb{P}(A_1 \cup \cdots \cup A_n)$.

Union of Events (Last Lecture)

Remarks:

1. If $A_1, \dots, A_n$ are mutually exclusive ($A_i \cap A_j = \emptyset$ for all $i \neq j$) events, then

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \mathbb{P}(A_i).$$

By additivity

2. **Union Bound:** For all events $A_1, \dots, A_n$ on the same probability space,

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} \mathbb{P}(A_i).$$

This inequality can be proved using induction.
This bound has many applications in Computer Science.

Union Bound Application

Theorem: If $n \le 2^{m/2}$ for $m \ge 3$, then there exists a 2-coloring of the edges of K_n such that it contains **no** monochromatic K_m subgraph (note: $n > m$).

Proof by Probabilistic Method: Prove existence by showing that there is positive probability that a randomly chosen object has the desired property.

Color each edge either blue or red with probability ½, independently.
Number of distinct K_m subgraphs of $K_n = \binom{n}{m}$.
Let E_i denote the event that the ith K_m subgraph is monochromatic.

blue or red

$$\mathbb{P}(E_i) = \frac{2}{2^{\binom{m}{2}}}$$

number of edges in the subgraph

$\mathbb{P}$(At least one K_m subgraph is monochromatic) =

$$\mathbb{P}\left(\bigcup_{i=1}^{\binom{n}{m}} E_i\right) \le \sum_{i=1}^{\binom{n}{m}} \mathbb{P}(E_i) = \binom{n}{m} \frac{2}{2^{\binom{m}{2}}} \le \frac{n^m}{m!} \frac{2}{2^{\binom{m}{2}}} \le \frac{2^{\frac{m^2}{2}}}{m!} \frac{2}{2^{\binom{m}{2}}} = \frac{2^{\frac{m}{2}+1}}{m!} < 1$$

Union bound — Since $n \le 2^{m/2}$ — for $m \ge 3$

$\mathbb{P}$(No K_m subgraph is monochromatic) $= 1 - \mathbb{P}$(At least one K_m subgraph is monochromatic)

> 0

□

Hashing

- Hashing maps an input from a large domain (e.g., strings, objects, vectors) into a fixed-size representation—typically an integer in a bounded range—via a hash function.
- **Hash function** $h: U \longrightarrow T$.
- Some "killer applications" of hashing (indispensable for performance, scalability, or even feasibility):
    - Databases (compression, deduplication, indexing)
    - Distributed systems (load balancing)
    - Bloom filters (space-efficient membership queries)
    - Cryptography (password storage, Blockchains)
    - Sketching & streaming algorithms (compact summaries)
    - Genomics & computational biology (e.g., Map DNA substrings to integers for fast lookup/counting)

Hashing

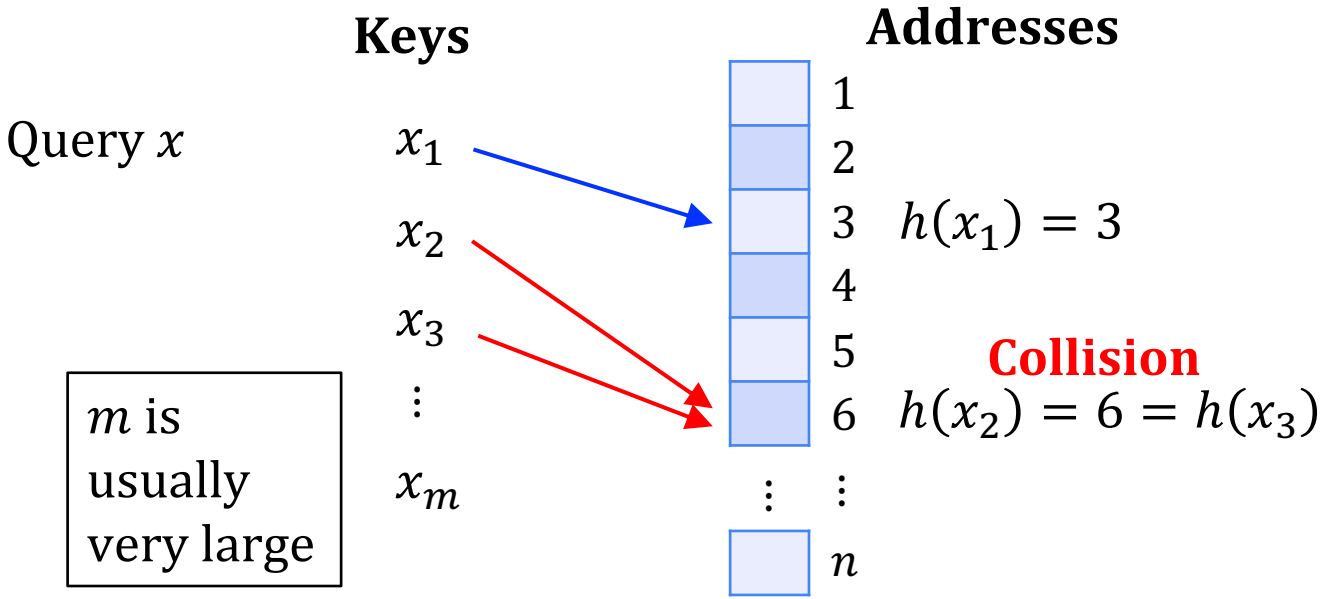
- **Hash function** $h: U \longrightarrow T$

Universe of keys

Table

Random Hash Function: map each key uniformly at random to n labeled bins, independently of other keys

- For $x \in U, h(x)$ is the location in T where data x is stored.



m is usually very large

Question: For given n and $\varepsilon > 0$, what is the **largest m** such that $\mathbb{P}(\text{collsion}) \leq \varepsilon$?

Question: How does that critical value of m scale with n and ε?

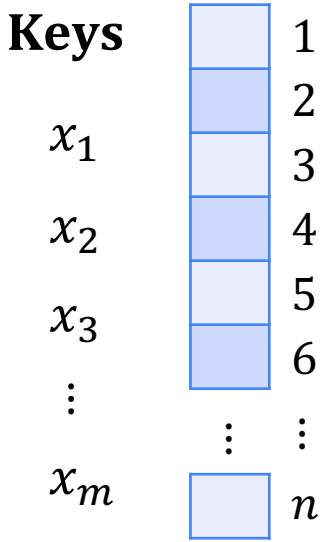
Hashing

higher m $\Rightarrow$ higher $\mathbb{P}(\text{collsion})$

Question: For given n and $\varepsilon > 0$, what is the **largest m** such that $\mathbb{P}(\text{collsion}) \leq \varepsilon$?

- Number of **pairs of keys** $= \binom{m}{2}$.
- For $i = 1, \dots, \binom{m}{2}$, let $C_i =$ event that pair i **collides.**
- $\mathbb{P}(C_i) = \frac{1}{n}$
- $C =$ at least one pair collides $= \bigcup_{i=1}^{\binom{m}{2}} C_i$
- $\mathbb{P}(C) = \mathbb{P}(\bigcup_{i=1}^{\binom{m}{2}} C_i) \leq \sum_{i=1}^{\binom{m}{2}} \mathbb{P}(C_i) = \binom{m}{2}\frac{1}{n} \approx \frac{m^2}{2}\frac{1}{n} \leq \varepsilon$

  by Union Bound
- So, $\mathbb{P}(C) \leq \varepsilon$ if $m \leq \sqrt{2\varepsilon n}$



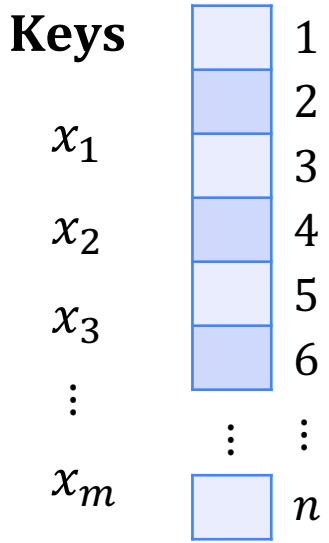
Hashing (a more refined analysis)

Question: For given n and $\varepsilon > 0$, what is the largest m such that $\mathbb{P}(\text{collsion}) \leq \varepsilon$?

- Let $\overline{C}$ = event of no collision
- We want $\mathbb{P}(\overline{C}) \geq 1 - \varepsilon$.

Recall the Birthday Paradox from Lecture 16.

- Independently throw m unlabeled balls u.a.r. into n labeled bins.
- $\mathbb{P}(\overline{C}) = \frac{n}{n}\left(\frac{n-1}{n}\right)\left(\frac{n-2}{n}\right)\cdots\left(\frac{n-m+1}{n}\right) = \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\cdots\left(1-\frac{m-1}{n}\right)$
- $\ln \mathbb{P}(\overline{C}) = \ln\left(1-\frac{1}{n}\right) + \ln\left(1-\frac{2}{n}\right) + \cdots + \ln\left(1-\frac{m-1}{n}\right)$.
- Recall Taylor series: $\ln(1-x) \approx -x$ for $x \ll 1$.
- $\ln \mathbb{P}(\overline{C}) \approx -\sum_{k=1}^{m-1} \frac{k}{n} = -\frac{m(m-1)}{2n} \approx -\frac{m^2}{2n}$.
- $\mathbb{P}(\overline{C}) \approx e^{-\frac{m^2}{2n}} \geq 1-\varepsilon \Rightarrow m \leq \sqrt{2n \ln\left(\frac{1}{1-\varepsilon}\right)}$
- For $\varepsilon \ll 1$, $\ln\left(\frac{1}{1-\varepsilon}\right) \approx \varepsilon$, which corresponds to the union bound.



Hashing (Comparison of Bounds)

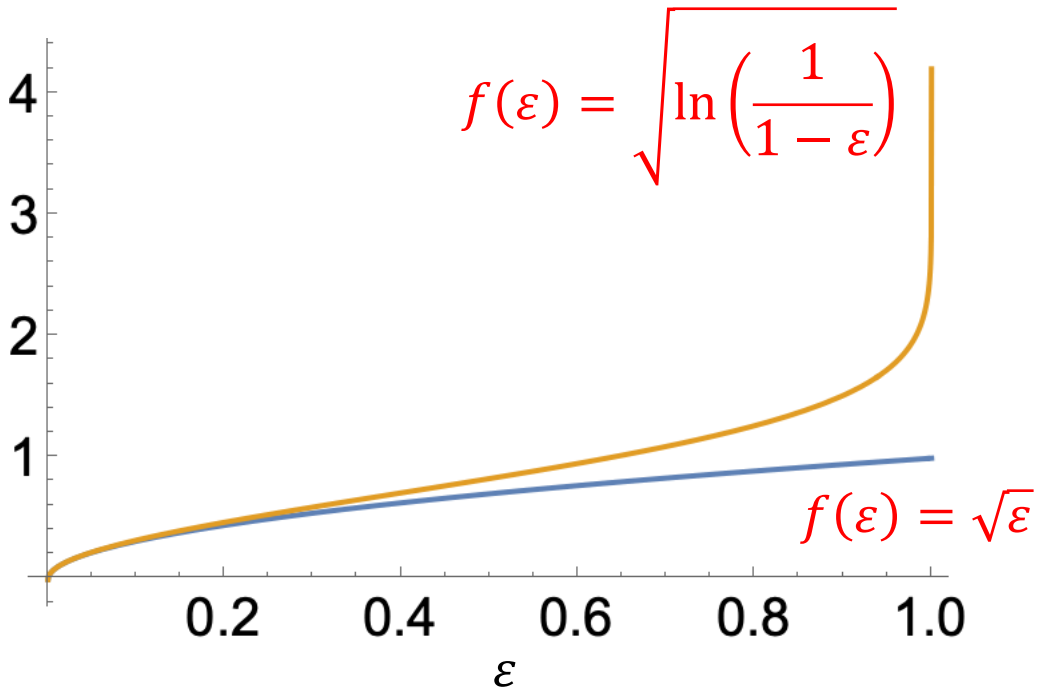
Question: For given n and $\varepsilon > 0$, what is the largest m such that $\mathbb{P}(\text{collsion}) \leq \varepsilon$?

(Union Bound)

$m \leq \sqrt{2n\varepsilon}$

(Birthday Paradox)

$$m \leq \sqrt{2n \ln\left(\frac{1}{1-\varepsilon}\right)}$$



Load Balancing

- Define L_i = load of processor i
- $A_k = \cup_{i=1}^{n}(L_i \geq k)$ and $\overline{A}_k = \cap_{i=1}^{n}(L_i < k)$

maximum load

All processors have less than k jobs allocated to them.

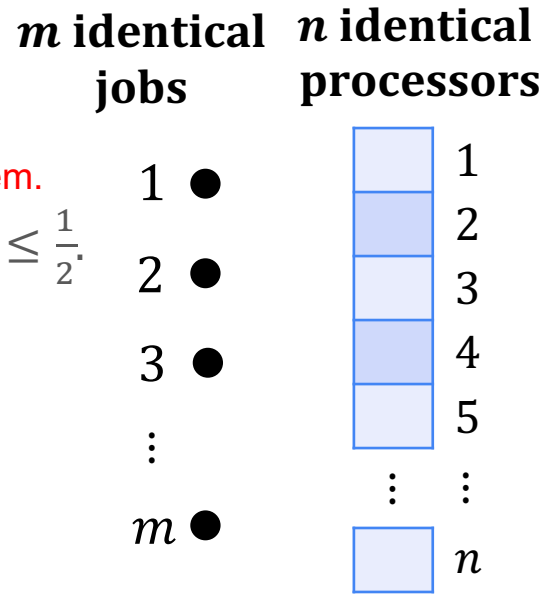
- Find the smallest k such that $\mathbb{P}(\overline{A}_k) \geq \frac{1}{2}$; equivalently $\mathbb{P}(A_k) \leq \frac{1}{2}$.
- $\mathbb{P}(A_k) = \mathbb{P}(\cup_{i=1}^{n}(L_i \geq k)) \leq \sum_{i=1}^{n} \mathbb{P}(L_i \geq k) = n\mathbb{P}(L_1 \geq k)$

- B_S = event that all jobs in $S \subseteq \{1, \dots, m\}$ land in bin 1.
- $\mathbb{P}(B_S) = \frac{1}{n^{|S|}}$.
- Then, $(L_1 \geq k) = \cup_{S \subseteq \{1,\dots,m\}:|S|=k} B_S$, so
- $\mathbb{P}(L_1 \geq k) = \mathbb{P}(\cup_{S \subseteq \{1,\dots,m\}:|S|=k} B_S) \leq \sum_{S \subseteq \{1,\dots,m\}:|S|=k} \mathbb{P}(B_S) = \binom{m}{k}\frac{1}{n^k}$

by Union Bound

- For $n = m$, $\mathbb{P}(A_k) \leq n\binom{n}{k}\frac{1}{n^k} \leq \frac{n}{k!}$

We want to choose k such that this is $\leq \frac{1}{2}$



Load Balancing

- $\frac{n}{k!} \leq \frac{1}{2}$
- Recall Stirling's formula from Lecture 14: For k large, $k! \sim \sqrt{2\pi k}\left(\frac{k}{e}\right)^k$
- Assuming large n, solve for k:

$$\ln(2\,n) = \ln\left(\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\right) \approx k \ln k$$

$$\ln\ln(2\,n) \approx \ln k + \ln(\ln k) \approx \ln k$$

With high probability (i.e., $\geq \frac{1}{2}$), the maximum load will be less than

$$k \approx \frac{\ln(2\,n)}{\ln\ln(2\,n)} \approx \frac{\ln(n)}{\ln\ln(n)}$$

(Note that this grows very slowly in n)

Coupon Collection

- n distinct coupons, 1 coupon per cereal box
- All coupon types are equally likely to be in any given cereal box
- Assume infinitely many cereal boxes so independence of sampling holds
- Collect all n distinct coupons to win
- A_i = Coupon i is **not** collected after m purchases.
- Again, think about throwing m unlabeled balls u.a.r. into n labeled bins.
- $\mathbb{P}(\cup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} \mathbb{P}(A_i) = n\left(1-\frac{1}{n}\right)^m \approx ne^{-m/n}$

Since $\left(1-\frac{1}{n}\right)^n \approx e^{-1}$ for large n

Suppose we want $\mathbb{P}(\cup_{i=1}^{n} A_i) \leq e^{-1} \approx 0.368$

By solving for m in $ne^{-m/n} = e^{-1}$, we conclude that if $m \geq n \ln n + n$, then $\mathbb{P}(\cup_{i=1}^{n} A_i) \leq e^{-1}$

Many problems in random processes and randomized algorithms (see CS174) can be modeled using the coupon collection problem.