PLAN

1. ROLLING HASH (RECAP)

2. PROBABILISTIC TOOLS
   - UNION BOUND
   - MARKOV'S INEQUALITY
   - LINEARITY OF EXPECTATION
   - AMPLIFICATION

3. BIRTHDAY PARADOX AND THE REVERSE PROBLEM

4. DNA PROBLEM

5. PATTERN MATCHING

**ROLLING HASH**

\[ h(S[i \ldots i+L]) = \left( S[i] \cdot b^{L-1} + S[i+1] \cdot b^{L-2} + \cdots + S[i+(L-1)] \right) \mod p \]

\[ h(S[i+1 \ldots i+L+1]) = \left( \left( b^{-1} \cdot b \right) + S[i+L] \right) \mod p \]

Can compute \((b^{-1})\mod p\) in \(O(L)\) time

Then \( h(S[i+1 \ldots i+L+1]) \) from \( h(S[i \ldots i+L]) \) in \(O(1)\) time
**TOOLS**

**UNION BOUND**

\[ B_1, \ldots, B_k \text{ - bad events} \]
\[ \Pr[\text{any bad event}] \leq \sum_{i=1}^{k} \Pr[B_i] \]

**MARKOV’S INEQUALITY**

\[ X \geq 0 \]
\[ \text{random variable} \]
\[ \alpha - \text{any positive real} \]
\[ \mathbb{E}[X] \geq \alpha \cdot \Pr[X \geq \alpha] \]

that is
\[ \Pr[X \geq \alpha] \leq \frac{\mathbb{E}[X]}{\alpha} \]

**Example:**

expected running time: 10s

probability runs longer than 100s: \( \leq \frac{10}{100} = \frac{1}{10} \)

**LINEARITY OF EXPECTATION**

\[ \mathbb{E}[aX + bY] = a \cdot \mathbb{E}[X] + b \cdot \mathbb{E}[Y] \]

**Example:**

run 5 times procedure A, expected time of A: 3s

run 2 times procedure B, expected time of B: 2s

total expected time: \( 5 \cdot 3s + 2 \cdot 2s = 19s \)
AMPLIFICATION

A = ALGORITHM THAT OUTPUTS YES/NO
CORRECT WITH PROBABILITY \( \geq \frac{2}{3} \)

RUN K TIMES AND RETURN MAJORITY

CAN SHOW: PROBABILITY OF ERROR \( \leq 2^{-\Omega(k)} \)

BIRTHDAY PARADOX

WE'RE SELECTING RANDOM NUMBERS FROM \( \{1, \ldots, N\} \)
WITH REPLACEMENT

HOW MANY SAMPLES TO DRAW THE SAME
NUMBER TWICE WITH CONSTANT PROBABILITY?

ANSWER: \( \Theta(\sqrt{N}) \)

WE'LL ONLY SHOW LOWER BOUND

SAY, WE PICK K NUMBERS: \( a_1, a_2, \ldots, a_k \)

\[
\Pr[a_i = a_j] = \frac{1}{N} \quad \text{(k\ \ \text{times})}
\]

\[
\Pr[\text{collision}] \leq \sum_{i \neq j} \Pr[a_i = a_j] = \frac{k(k-1)}{2N} \cdot \frac{1}{N}
\]
Example 1: Want
Then
\[ \frac{1}{2} \leq \frac{k(k-1)}{2} \cdot \frac{1}{N} \]
\[ N \leq k(k-1) \]
So
\[ k^2 \geq N \]
\[ k \geq \sqrt{N} \]

Example 2: \( n \) elements hashed into \( \{1 \ldots 100n^2\} \)
uniformly
\[ \Pr[\text{collision}] \leq \frac{n(n-1)}{2} \cdot \frac{1}{100n^2} = \frac{1}{200} \]

DNA Problem
(uniform hashing)

Attempt 1: algorithm from the last lecture: create \( N=O(n^2) \)-size array
expected number of collisions in each iteration
\[ \leq \binom{n}{2} \cdot \frac{1}{N} = O(1) \]
rolling hash: can compute all hash values in \( O(n) \) time
expected time for comparing strings: \( O(1) \cdot O(n)=O(n) \)
number of iterations: \( O(\log n) \)
total expected time: \( O(n \log n) + O(n^2) \) initialization
ATTEMPT 2: HASH INTO A TABLE
OF SIZE $\Theta(n)$ BUT COMPARE 
FIRST SIGNATURES FROM $\{1, \ldots, n^2\}$

WHEN INSERTING NEW SUBSTRING, EXPECTED NUMBER
OF SIGNATURE COMPARISONS = $O(1)$

EXPECTED TOTAL NUMBER OF STRING COMPARISONS
STILL $O(1)$

EXPECTED RUNNING TIME: $O(n) + O(n) \cdot O(1) + O(1) \cdot O(n) = O(n)$

ALL $O(\log n)$ ITERATIONS: $O(n \log n)$

PATTERN MATCHING
FIND AN OCCURRENCE OF PATTERN $x$
IN TEXT $y$

NAIVE SOLUTION: $O(|x| \cdot |y|)$
WANT $O(|y|)$
1. Compute rolling hash for all subwords of $y$ of length $|x|$

2. Run naive check for subwords $y'$ such that $h(y') = h(x)$ until you find occurrence of $x$

If probability that $h(y') = h(x)$ is $O(1/|x|)$ for any $y'$, then expected running time:

$$O(|y|) + O(|y|) \cdot O(1/|x|) \cdot O(|x|) = O(|y|)$$

rolling hash

This algorithm known as Robin-Karp algorithm

Our rolling hash invented for this algorithm