1 Review

- Hashing-based Algorithms
  - Hash tables
  - Universal hashing
  - Uniform random hashing analysis
  - Bloom filter
  - Quotient filter

- Probability
  - Random Variable
  - Expectation
  - Linearity of expectation
  - Union bounds
  - Chernoff bounds

\[ X \leq \mu + O\left(\sqrt{\mu \log n}\right) \quad \text{w.h.p.} \quad \mu = \Omega(\log n) \]

- Approximations

\[ \left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e}, \quad \left(1 + \frac{1}{n}\right)^n \approx e \]

\[ n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \]

- with high probability (w.h.p.)

\[ \forall \alpha, \exists C_\alpha, c, N, \ s.t. \ \forall n > N, \ P\left[X > c(\cdot)\right] > 1 - \frac{C_\alpha}{n^\alpha} \]
Divide and Conquer

- Sorting
  * merge sort
  * quick sort (randomized & deterministic)
- Median finding (i.e. kth)
- Recursive relations
- Master Theorem
- Matrix Multiplication (Strassens)
- Nearest pair of points

2 The Collision Problem

Problem 1. Given n time slots, how many time slots must be chosen in order to get 2 w.h.p.?

⇒ collision probability
⇒ throw k balls, what’s that chance that ≥ 2 balls land in same bin?

Solution:

\[
P[\text{all } k \text{ balls go in different bins}] = \frac{n}{n} \cdot \frac{n-1}{n} \cdots \frac{n-k+1}{n}
\]

\[
= \frac{n!}{(n-k)!n^k} = \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{\sqrt{2\pi (n-k)} \left(\frac{n-k}{e}\right)^{n-k} n^k}
\]

\[
= \left(\frac{n}{n-k}\right)^{n-k+\frac{1}{2}} \cdot \left(\frac{n-k}{n}\right)^{\frac{k}{2}} \cdot \frac{1}{e^k} = \frac{1}{e^k} \left(1 - \frac{k}{n}\right)^{k-n-\frac{1}{2}}
\]

\[
= e^{-\frac{k^2}{2n} + \frac{k}{2n}}
\]

The goal is to prove:

Given α, find k, s.t. \(e^{-\frac{k^2}{2n} + \frac{k}{2n}} \leq \frac{1}{n^\alpha}\)

⇒ \(-\frac{k^2}{n} + \frac{k}{2n} \leq -\alpha \ln n\)

⇒ \(k^2 - \frac{k}{2} \geq \alpha n \ln n\)

⇒ \(k \geq \sqrt{\alpha n \ln n}\)

Theorem 2. Throw \(\theta (\sqrt{n \log n})\) balls into n bins will result a collision w.h.p.
3 3 SAT

The problem can be summarized as:

- $n$ variables $X_1, \ldots, X_n$
- $k$ clauses $C_1, \ldots, C_n$, each $C_i$ is union of three variables, example $C_1 = X_1 \cup X_7 \cup \overline{X}_{35}$
- check $C_1 \cap C_2 \cap \cdots \cap C_n$

**Example 3.** Given the result of $(X_1 \cup X_7 \cup \overline{X}_{35}) \cap (X_1 \cup X_7 \cup \overline{X}_{37})$, find possible solution of $\{X_i\}$.

In order to solve the problem, suppose each clause only uses variables at most 10 apart.

Sort clause by index of lowest variable:

\[
\begin{array}{c}
C_1, \ldots, C_k, \\
X_1, \ldots, X_l \\
C_{k+1}, \ldots, C_k \\
X_{l-9}, \ldots, X_n
\end{array}
\]

And the idea of this algorithm is:

Compute set of values for $X_{l-9}, \ldots, X_l$ for which there is a solution to $C_1, \ldots, C_{k/2}$ and $C_{k/2+1}, \ldots, C_k$ agreeing on $X_{l-9}, \ldots, X_l$. Then check intersection.

The algorithm can be implemented as:

\[
\text{Find Solution} \left(C_1, \ldots, C_k\right) \\
\text{(this algorithm returns } \{(X_1, \ldots, X_{10}, X_{n-9}, \ldots, X_n)\}\text{)}
\]

if $n \leq 20$

brite force

else

$S_1 = \text{Find Solution} \left(C_1, \ldots, C_{k/2}\right)$

$S_2 = \text{Find Solution} \left(C_{k/2+1}, \ldots, C_k\right)$

return $S_1 \bowtie S_2$

The operation of this algorithm is

\[
T(n) = 2T\left(\frac{n}{2}\right) + O(1)
\]

thus, according to the Master theorem

\[
T(n) = O(n).
\]