CS 630 – Fall 2024 – Lab 7
Oct 30, 2024

1. Hashing

Suppose we have a hash function defined as $h(x) = (3x + 5) \mod n$ where n equals to the hash table size, and we have following elements $[42, 56, 21, 17, 29, 18, 55, 80, 20, 1]$.

1. Supporse we have a hash table of size 10, what is the resulting hash table.

   Solution:
   The hash-table will look like this: $[[55], [42], [29], [56], [], [80, 20], [17], [], [21, 1], [18]]$

2. What is the ideal hash table size if we want to have a collision rate for each cell that is lower than 10% with 10 elements to insert?

   Solution:
   This is the same calculation as the birthday paradox. There exists $10 \times 9/2 = 45$ possible parings between the 10 elements (suppose they are unique). Then the probability that each pair has $\frac{n-1}{n}$ chance of not colliding with uniform hashing. Therefore we get:

   $$\Pr(\text{at least one pair share the same hashing}) = 1 - (\frac{n-1}{n})^{45} = 10\%$$

   $$n \approx 427.61$$

2. Markov v.s. Chernoff bounds

Let X be a random variable representing the number of heads obtained when flipping n independent coins, each with a probability p of showing heads. We know that X follows a binomial distribution, $X \sim \text{Binomial}(n, p)$, with $E[X] = np$ and $Var[X] = np(1-p)$

1. Calculate an upper bound for $P(X \geq a)$ where $a > E[X]$ using Markov's Inequality.

Solution:

$$P(X \geq a) \leq \frac{np}{a}$$

2. Calculate an upper bound for $P(X \geq a)$ where $a > E[X]$ using Chebyshev's Inequality.

Solution:

$$\begin{aligned} P(X \geq a) &= P(X - np \geq a - np) \\ &\leq P(|X - np| \geq a - np) \\ &= P\left(|X - np| \geq \frac{a - np}{\sqrt{(np(1-p))}}\sqrt{(np(1-p))}\right) \\ &\leq \frac{np(1-p)}{(a-np)^2} \end{aligned}$$

3. Calculate an upper bound for $P(X \geq a)$ where $a > E[X]$ using Chernoff Bound.

Solution:

$$\begin{aligned} P(X \geq a) &= P\left(X \geq \frac{a}{np} np\right) \\ &= P\left(X \geq \Big(1 + \frac{a - np}{np}\Big) np\right) \\ &\leq \Big(\frac{e^{\frac{a-np}{np}}}{(a/np)^{a/np}}\Big)^{np} \end{aligned}$$

3. Better chance

Given a function with randomized output that is correct with probability of 0.6 with fixed time c, write an algorithm that can produce a correct result with probability higher than 0.9. Create the algorithm and analyze how much time the algorithm is expected to take.

Solution:
We repeat the function n time and take the answer that is the majority. Let X be number of wrong answers. so $X \sim \text{Binomial}(n, 0.4)$, so we have $E[X] = 0.4n$ and $Var[X] = 0.24n$. So the probability of our algorithm returns an error can be bounded by Chebyshev's Inequality.

$$\begin{aligned} P(X \geq 0.5n) &= P(X - 0.4n \geq 0.1n) \\ &\leq P(|X - 0.4n| \geq 0.1n) \\ &= P(|X - 0.4n| \geq \frac{0.1\sqrt{n}}{\sqrt{0.24}}\sqrt{0.24n}) \\ &\leq \frac{24}{n} \end{aligned}$$

To get success rate of 0.9 we can set $n = 240$ so that the fail rate is ≤ 0.1 so we get $P(correct) \geq 0.9$

4. Las Vegas and Monte Carlo Algorithm

A Monte Carlo algorithm is a type of randomized algorithm that may produce incorrect results with a small probability but typically runs in deterministic time. In contrast, a Las Vegas algorithm always produces a correct result, but its running time may vary.

1. Given a Monte Carlo algorithm $MC(x)$ that has a probability p of outputting the correct answer with $O(1)$ time complexity and a verifier $V(x)$ that verifies the output in $O(1)$ time. Write a Las Vegas Algorithm $LV(x)$ and give a expected time complexity analysis.

   Solution:
   Repeat calling $MC(x)$ and verifying the result with $V(x)$, stop when $V(x)$ returns true. The number of repetition is a geometric distribution with probability of success p. The expected runtime is $O(1/p)$

2. Given a Las Vegas algorithm $LV(x)$ with a runtime of $T(x)$. Write a Monte Carlo algorithm $MC(x)$ with a constant time complexity.

   Solution:
   Specify some constant time T, run $LV(x)$, early stop if $LV(x)$ returns an output by T, return the output, else return nothing. This will have $O(1)$ time complexity.