COMPSCI 330: Design and Analysis of Algorithms	October 28, 2014
Lecture # 17	
Lecturer: Debmalya Panigrahi	*Scribe: Roger Zou*

1 Overview

We introduce the topic of **randomized algorithms**, beginning with two examples: the coupon collector problem and shared resource contention. Along the way, we show techniques in the analysis of randomized algorithms, through linearity of expectation, independence, and useful mathematical facts. Our main goal for this class will be bounding the runtime of randomized algorithms for both problems, in expectation.

2 Introduction

Pros and Cons of Randomized Algorithms

Pros: Simplifies algorithm substantially. Also, for some problems, the best algorithms are randomized; we either don't know any complementary deterministic algorithm, or that a randomized algorithm is provably better than its deterministic complement.
Cons: Randomized doesn't seem to help much with improving complexity.

Two types of Randomized Algorithms

Monte Carlo Algorithm: Randomized algorithm that gives the correct answer with probability $1 - o(1)$ ("high probability"), but the runtime bounds hold deterministically.
Las Vegas Algorithm: Randomized algorithm that always gives the correct answer, but the runtime bounds hold in expectation.

3 Coupon Collector Problem

Problem Statement

There are n coupons. In each draw, a coupon is drawn uniformly at random (uar). How many coupons do I need to draw to get all n coupons (in expectation)?

Solution

Let X_i be the random variable representing number of draws between the $(i-1)^{th}$ distinct coupon and the i^{th} distinct coupon.

Definition 1. *Let X be a random variable reflecting the number of independent Bernoulli trials (with probability p of success) needed to observe the first success. Then the probability that the first success occurs at the k^{th} trial follows a Geometric distribution with*

$$Pr(X = k) = p(1-p)^{k-1}$$

Let p_i be the probability of drawing an unseen coupon at the i-th draw. Then

$$p_i = \frac{n-i+1}{n}$$

Let q_i be the probability of drawing a seen coupon. Then

$$q_i = \frac{i-1}{n}$$

Thus the probability of taking k draws to get the i^{th} unseen coupon is as follows, which follows from independence between draws.

$$Pr(X_i = k) = p_i q_i^{k-1}$$

Thus X_i is a geometric random variable.
Let X be the number of coupons drawn until all n distinct coupons show up, which is what we want to find. First, note that $X = \sum_{i=1}^{n} X_i$. By linearity of expectation, it follows that $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i]$.

Our next step is to find the expectation of X_i.

$$\begin{aligned}
\mathbb{E}[X_i] &= \sum_{k=1}^{\infty} k\,Pr[X_i = k] \\
&= \sum_{k=1}^{\infty} k\,q_i^{k-1} p_i \\
&= \frac{1}{p_i} \\
&= \frac{n}{n-i+1}
\end{aligned}$$

From this we can find the expected number of coupons drawn until all n distinct coupons show up, which is simply the sum of all $\mathbb{E}[X_i]$, for all $i = 1, \ldots, n$.

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=1}^{n} \mathbb{E}[X_i] \\
&= \sum_{i=1}^{n} \frac{n}{n-i+1} \\
&= n\left(\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} + \frac{1}{1}\right) \\
&= \Theta(n \log n)
\end{aligned}$$

Note that the partial sum of the harmonic series has logarithmic growth.

4 Shared Resource Contention

Problem Statement

We have n processes $p_1, \ldots, p_n$, all trying to access a shared resource S. Each process tosses a random coin with success probability $1/n$. If the coin returns true for process p_i, then p_i tries to access the resource.

If there is no **contention** with other processes (p_i is only process that tries to access the resource), then p_i accesses the resource. If there is contention, then no process accesses the resource. Our goal is to determine how many rounds are required for all processes to access the resource (in expectation).

Solution

Let X_i be the number of rounds between the $(i-1)^{th}$ and i^{th} process accessing S. Let X be the total number of rounds needed, and we wish to find $\mathbb{E}[X]$.
Again, note that $Pr(X_i = k) = q_i^{k-1} p_i$ and so $\mathbb{E}[X_i] = \frac{1}{p}$. Here we will leverage the useful fact that

Fact 1.

$$\lim_{x\to\infty}\left(1-\frac{1}{n}\right)^n = \frac{1}{e} = \Theta(1)$$

In addition,

$$\lim_{x\to\infty}\left(1-\frac{1}{n}\right)^{n-1} = \frac{1}{e} = \Theta(1)$$

Let p_i be the probability that a new process access the resource in any round, and let its complement be $q_i = 1 - p_i$. Then

$$\begin{aligned} p_i &= (n-i+1)\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \\ &= \frac{n-i+1}{n}\Theta(1) \end{aligned}$$

Note that $\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$ is the probability that the i-th process returns true with the rest being false, and $(n-i+1)$ is the number of possible processes. The second step substitutes $\Theta(1)$ based on Fact 1. Then

$$\begin{aligned} \mathbb{E}[X_i] &= \frac{1}{p_i} \\ &= \frac{n}{n-i+1}\Theta(1) \\ &= \Theta(\log n) \end{aligned}$$

Again, similar to the coupon collector problem, $\mathbb{E}[X] = \Theta(n\log n)$ by linearity of expectation.

5 Summary

In this lecture we introduced Randomized Algorithms, with a brief overview of two classes of randomized algorithms: Monte Carlo and Las Vegas Algorithms. We then discussed two applications of Las Vegas algorithms: the Coupon Collector Problem and the Shared Resource Contention Problem. We used basic probability and useful facts from mathematics to prove the number of iterations it takes for the algorithm to complete, in expectation.