Lecture Notes CS:5350 Design and Analysis of Algorithms
Lecture 17: October 22, 2019
Scribe: Jeehan Malik

1 Randomized Quicksort

In this lecture, the quicksort algorithm is analyzed and its **expected** runtime is proved by using the probabilistic concepts of **random variables** and **linearity of expectation**.

A key part of the randomized quicksort algorithm (from last class) was:

```
i <- index chosen uniformly at random from [1...|L|]
swap(L, 1, i)
for j <- 2 to |L| do:
    if L[j] <= L[i]:
        append L[j] to L1
    else:
        append L[j] to L2
```

Observation 1 RANDOMIZED QUICKSORT *The running time of this algorithm is proportional to the number of comparisons we make because every iteration of the loop makes one comparison and the rest of the code runs in constant time. Therefore, when we are thinking about running time for quicksort, it suffices to count the number of comparisons made.*

Let X = random variable denoting the number of comparisons. We will now prove the following theorem.

Theorem 1 RANDOMIZED QUICKSORT *can be solved with expected $O(n \log n)$ comparisons. That is,* $\mathbb{E}[X] = \mathcal{O}(n \log n)$

Main Idea: To prove this, we will express the random variable X as a sum of some decomposed, simpler random variables. It will be easier to figure out the expectations of those random variables. Then, we will use linearity of expectation to find the expectation for X.

1.1 Decomposition of X

Let the input list be (x_1, x_2 , ... , x_n) and let the sequence (y_1, y_2 , ... , y_n) be a sorted version of L. For our analysis we will consider the sorted sequence.

Let X_{ij} for $1 \leq i < j \leq n$ denote **indicator random variables**, where i and j refer to indices in the sorted array. **Indicator random variables** refer to binary (having values 0 or 1) random variables indicating whether an event has occurred or not. So, X_{ij} indicates whether y_i and y_j have been compared.

$$X_{ij} = \begin{cases} 1 & \text{if } y_i \text{ and } y_j \text{ are compared} \\ 0 & \text{otherwise} \end{cases}$$

Note that y_i and y_j can only be compared once during the course of the algorithm. This means that X_{ij} is also the sum of the number of comparisons between y_i and y_j.

We can now decompose X into many indicator random variables, because the total number of comparisons made by the algorithm is the sum of indicator variables across all possible values of y_i and y_j

The main decomposition step is:

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

1.2 Using Linearity of Expectation

$$\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}]$$

By linearity of expectation:

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[X_{ij}]$$

By definition of expectation:

$$\begin{aligned} \mathbb{E}[X_{ij}] &= 1 \cdot Pr(X_{ij} = 1) + 0 \cdot Pr(X_{ij} = 0) \\ &= Pr(X_{ij} = 1) \\ \therefore \mathbb{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr(X_{ij} = 1) \end{aligned}$$

Now we have to determine what $Pr(X_{ij} = 1)$ is, in order to figure out the expectation of X.

1.3 Determining Probability

In order for y_i and y_j to be compared, one of them has to be pivot. Alternately, what would prevent them from being compared is if they end up in different lists. y_i and y_j would end up in different lists if a random pivot is picked with index between i and j. Figure 1 shows a visualization of this.

From the figure, we can see that if the pivot is picked from the area indicated as k, the situation of whether or not y_i and y_j are compared, would be resolved.
At some point in the algorithm, a pivot will be chosen from the area between y_i and y_j. We can therefore see that:

Picture:

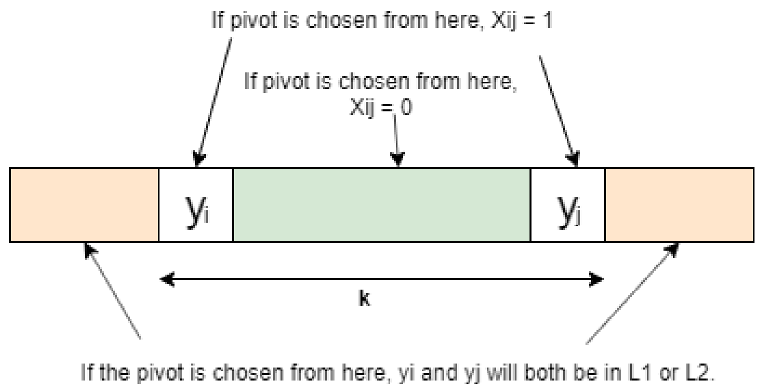

Figure 1: A visual showing the indexes of y_i and y_j in an array.

$$Pr(X_{ij} = 1) = \frac{2}{j - i + 1}$$

So,

$$\mathbb{E}[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}$$

Let $k = j - i + 1$ (the length of sequence between y_i and y_j)

$$
\begin{aligned}
&= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&= \sum_{k=2}^{n} \sum_{i=1}^{n-k+1} \frac{2}{k} \\
&= \sum_{k=2}^{n} \frac{2}{k} \cdot (n-k+1) \\
&= 2n \sum_{k=2}^{n} \frac{1}{k} - \sum_{k=2}^{n} 2 + \sum_{k=2}^{n} \frac{2}{k} \\
&= 2n(H_n - 1) - 2(n-1) + 2(H_n - 1)
\end{aligned}
$$

where H_n is the n^{th} Harmonic Number in the sequence $1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$

We know that $H_n = \Theta(\log n)$

$$\therefore \mathbb{E}[X] = \mathcal{O}(n \log n)$$

1.4 Other ways of analyzing randomized quicksort

This is one way to analyze randomized quicksort. The traditional way to analyze quicksort is in terms of recurrences. Since partitioning of a list is randomized, we can apply expectation over recurrence.

$$T(|L|) = T(L_1) + T(L_2) + O(n)$$

When we take the expectation over both sides of this equation, with some work we can show that this reduces to:

$$\mathbb{E}[T(n)] = 2 \cdot \mathbb{E}[T(n)] + O(n)$$

After this, we can solve this equation as usual.

1.5 Final Note

Think about the following modification in our algorithm: Replace a randomized choice of pivot by the Balanced Partition Monte Carlo Algorithm. For example, take L_1 and L_2 and if they are unbalanced, the algorithm gives up. This modification would guarantee that our running time is not a random variable, it is deterministic. But, now the algorithm will sometimes return "error" and so we need to analyze the error probability of this algorithm.

2 Coupon Collector's Problem

This is another problem that provides an illustration of the linearity of expectation. We have cereal boxes, each of which has one of n coupons (1, 2,. . . , n) chosen uniformly at random.

Problem: How many cereal boxes do you need to buy in order to have all n coupons?

Let X = random variable denoting the number of cereal boxes purchased to get n coupons.
We are interested in what $\mathbb{E}[X]$ is. We can use the same approach of decomposing X into simpler random variables followed by using the linearity of expectation, as we did with quicksort.

Picture:

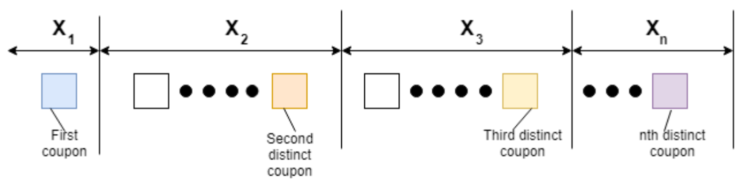

Figure 2: A pictorial representation of the decomposition of coupons in the Coupon Collector's problem.

Let X_i denote the number of cereal boxes purchased to get the i^{th} coupon after i-1 coupons have been obtained. So $X_1 = 1$ and X_2, X_3 , ... , X_n are illustrated in Figure 2. Now note that:

$$X = \sum_{i=1}^{n} X_i$$

We will proceed by finding expectations of each of these sequences.