All About Quicksort

I. M. Student
Department of Mathematics and Computer Science
Denison University
Granville, Ohio 43023

February 25, 2004

1 Introduction

The Quicksort algorithm was invented in 1962 by C. A. R. Hoare.[1] It is generally considered to be the fastest sorting algorithm in practice when the range of elements to be sorted is unknown.

In this paper, we will present the Randomized Quicksort algorithm and prove that its expected running time is in $O(n \log n)$.

2 The Quicksort Algorithm

We will present the Randomized Quicksort algorithm below. This pseudocode is modeled after that in the book by Cormen, et al. [2].

```
Partition(A, p, r)
   x <- A[p]
   i <- p - 1
   j <- r + 1
   while true do
      repeat
         j <- j + 1
      until A[j] <= x

      repeat
         i <- i + 1
      until A[i] >= x

      if i < j then
         exchange(A[i], A[j])
      else
         return j
```

```
Randomized-Partition(A, p, r)
   i <- Random(p, r)
   exchange(A[r], A[i])
   return Partition(A, p, r)

Randomized-Quicksort(A, p, r)
   if p < r then
      q <- Randomized-Partition(A, p, r)
      Randomized-Quicksort(A, p, q - 1)
      Randomized-Quicksort(A, q + 1, r)
```

3 Expected Running Time of Randomized Quicksort

Let $T(n)$ be the running time of the randomized quicksort algorithm on an array of n elements. Then, since all the work in quicksort is done in the PARTITION routine, we can say that

$$T(n) = O(C + X)$$

where C is the number of calls to PARTITION (the call itself requires constant time) and X is the total number of comparisons performed in the entire algorithm (in all the calls to PARTITION). Since this is a randomized algorithm, we will eventually want to find the expected running time $E[T(n)]$.

First, notice that after every call to PARTITION, the pivot used in that call cannot be used as the pivot in a subsequent partition, since it is not included in either of the recursive calls to QUICKSORT. This implies that each of the n elements can be a pivot at most once. Therefore, there can be at most n calls to PARTITION and hence,

$$C \leq n\,. \tag{1}$$

Now, let $z_1, z_2, \ldots, z_n$ denote the elements in the list to be sorted, where z_i is the i^{th} smallest element. Also, let $Z_{ij} = \{z_i, z_{i+1}, \ldots, z_j\}$.

We next define an indicator random variable X_{ij} as follows:

$$X_{ij} = \begin{cases} 1 & \text{if } z_i \text{ is compared to } z_j \\ 0 & \text{otherwise} \end{cases}$$

Then, since elements are only ever compared to the pivot and each pivot is used at most once, we know that the total number of comparisons in QUICKSORT is the sum of all the X_{ij}'s. This quantity is

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

The summation simply accounts for all possible X_{ij}'s without including X_{ii}'s and X_{ji}'s, since these would be redundant.

To get the expected (average) running time, we need the expected value of X, denoted $E[X]$:

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X_{ij}\right] \\ &= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} E\left[X_{ij}\right] \end{aligned} \tag{2}$$

By the definition of expected value,

$$E\left[X_{ij}\right] = 1 \cdot P(z_i \text{ is compared to } z_j) + 0 \cdot P(z_i \text{ is not compared to } z_j) = P(z_i \text{ is compared to } z_j)\,.$$

Therefore, substituting into (2), we have

$$E[X] = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} P(z_i \text{ is compared to } z_j)\,. \tag{3}$$

Now we observe two facts about QUICKSORT:

1. Once a pivot x is chosen where $z_i < x < z_j$, z_i and z_j can never be compared again.
2. If z_i (or z_j) is chosen as a pivot before any other element in Z_{ij}, then z_i (or z_j) will be compared to all other elements in Z_{ij} except itself.

From these two statements, we can conclude that z_i and z_j are compared if and only if the first element to be chosen as a pivot in each Z_{ij} is either z_i or z_j. (If another element is chosen as a pivot first then, since this element must be between z_i and z_j, z_i and z_j will not be compared (by observation 1 above)). Therefore,

$$\begin{aligned} P(z_i \text{ is compared to } z_j) &= P(z_i \text{ is chosen as the first pivot from } Z_{ij} \text{ or } z_j \text{ is chosen as the first pivot from } Z_{ij}) \\ &= P(z_i \text{ is chosen as the first pivot from } Z_{ij}) + P(z_j \text{ is chosen as the first pivot from } Z_{ij}) \\ &= \frac{1}{j-i+1} + \frac{1}{j-i+1} \\ &= \frac{2}{j-i+1}\,. \end{aligned}$$

Now, continuing from (3), we have

$$\begin{aligned}
E[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P(z_i \text{ is compared to } z_j) \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} && \text{by substituting } k \text{ for } j-i \\
&= \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k} \\
&< 2 \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{1}{k} \\
&\leq 2 \sum_{i=1}^{n-1} (\ln n + 1) && \text{since } \sum_{k=1}^{n} 1/k \leq \ln n + 1 \\
&= \Theta(n \lg n)\,. && (4)
\end{aligned}$$

Lastly, using (1) and (4), we have

$$\begin{aligned}
E[T(n)] &= E[O(C+X)] \\
&= O(n + E[X]) \\
&= O(n + n \lg n) \\
&= O(n \lg n)\,.
\end{aligned}$$

In other words, the expected running time of the randomized quicksort algorithm is $O(n \lg n)$.

4 Conclusions

The Quicksort algorithm is a very efficient sorting algorithm in the average case. In fact, it is optimal in the average case because it is well known that $\Omega(n \log n)$ steps are required to sort any list using a comparison sort.

References

[1] C. A. R. Hoare. Quicksort. *Computer Journal,* 5(1):10–15, 1962.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, MIT Press and McGraw-Hill, second edition, 2001.