Average Case Quick Sort Analysis

Now, to the average case running time. This is certainly difficult to ascertain because we could get any sort of partition. We will assume that each possible partition (0 and n-1, 1 and n-2, 2 and n-3, etc.) is equally likely. One way to work out the math is as follows:

Assume that you run Quick Sort n times. In doing so, since there are n possible partitions, each equally likely, on average, we have each partition occur once. So we have the following recurrence relation:

$$nT(n) = T(0)+T(n-1)+T(1)+T(n-2)+...+T(n-1)+T(0) + n*n$$
$$nT(n) = 2[T(1)+T(2)+...T(n-1)] + n^2$$

(The n is for the work done by the partition method, simplified from O(n) to make the analysis easier.)

Now, plug in n-1 in the equation above to get the following one:

$$(n-1)T(n-1) = 2[T(1)+T(2)+...T(n-2)] + (n-1)^2$$

Subtracting these two equations we get:

$$nT(n) - (n-1)T(n-1) = 2T(n-1) + 2n - 1$$
$$nT(n) = (n+1)T(n-1) + (2n - 1)$$
$$T(n) = [(n+1)/n]T(n-1) + (2n - 1)/n$$

Since we are only trying to do an approximate analysis, we will drop the -1 at the end of this equation. Dividing by n+1 yields:

$$T(n)/(n+1) = T(n-1)/n + 2/(n+1)$$

Now, plug in different values of n into this recurrence to form several equations:

$T(n)/(n+1) = T(n-1)/n + 2/(n+1)$
$T(n-1)/(n) = T(n-2)/(n-1) + 2/(n)$
$T(n-2)/(n-1) = T(n-3)/(n-2) + 2/(n-1)$
...
$T(2)/3 = T(1)/2 + 2/1$

Now, adding all of these equations up reveals many identical terms on both sides. In fact, after cancelling identical terms, we are left with:

$T(n)/(n+1) = T(1)/2 + 2[1/1 + 1/2 + 1/3 + ... + 1/(n+1)]$

The sum on the right hand side of the equation is a harmonic number. The nth harmonic number(H_n) is defined as $1 + 1/2 + 1/3 + ... 1/n$.

Through some calculus, it can be shown that $H_n \sim \ln n$. (ln is the natural log. It is a logarithm with the base e. $e \sim 2.718282$.)

Now, we have:

$T(n)/(n+1) \sim 1/2 + 2\ln n$
$T(n) \sim n(\ln n)$, simplifying a bit.

Thus, even in the average case for Quick Sort, we find that $T(n) = O(n \log n)$.

Note, in order analysis, any function of the form $\log_b n = O(\log_c n)$, for all positive constants b and c, greater than 1.

Average Case Analysis for Quick Select

In Quick Select, we are looking for the k^{th} smallest item in an array of size n. We always to the same first step:

1. Partition the array.

Then, we'll take one of three actions:

(a) We got lucky and found the k^{th} item, return it.

(b) The item must be on the left, let this be size x. We must recursively look for the item in an array of size x.

(c) The item must be on the right, let this be size n – 1 – x. We must recursively look for the item in an array of size n – 1 – x.

The probability of (a) occurring is 1/n.

The probability of (b) occurring is x/n.

The probability of (c) occurring is (n – 1 – x)/n.

Of course, the probability x = 0 is 1/n, the probability x = 1 is 1/n, the probability x = 2 is 1/n, …, the probability x = n – 1 is 1/n.

Putting this all together, let T(n) be the average case run time of Quick Select. Then we can put together the following recurrence relation that T(n) satisfies:

$$\begin{aligned} T(n) = n + \frac{1}{n}(1) + \frac{1}{n}\left(\frac{0}{n}T(0) + \frac{(n-1)}{n}T(n-1)\right) \\ + \frac{1}{n}\left(\frac{1}{n}T(1) + \frac{(n-2)}{n}T(n-2)\right) + \\ + \frac{1}{n}\left(\frac{2}{n}T(2) + \frac{(n-3)}{n}T(n-3)\right) + \cdots \\ + \frac{1}{n}\left(\frac{(n-1)}{n}T(n-1) + \frac{0}{n}T(0)\right) + \end{aligned}$$

This is a lot to unpack. Let's look at teach term.

n **→ first term represents Partition time.**

$\frac{1}{n}(1)$ **→ represents our return statement in the lucky case that The partition element IS the kth smallest**

$\frac{1}{n}\left(\frac{0}{n}T(0) + \frac{(n-1)}{n}T(n-1)\right)$ **→ This is the chance that the partition element is the smallest, so the left of partition is 0 elements and the right of partition is n-1 elements. The probability that the kth element is on the left is 0/n and the probability that it's on the right is (n-1)/n. Note that the 1/n chance that we got the element was already included in the previous term.**

In general any term of the form

$\boldsymbol{\frac{1}{n}\left(\frac{x}{n}T(x)+\frac{(n-1-x)}{n}T(n-1-x)\right)}$ → **represents the contribution to T(n) when the partition split is x elements on the left and n – 1 – x elements on the right.**

Writing this in a formalized summation, we get:

$$\boldsymbol{T(n)=n+\frac{1}{n}(1)+\frac{1}{n}\sum_{i=0}^{n-1}(\frac{i}{n}T(i)+\frac{(n-1-i)}{n}T(n-1-i))}$$

Notice that the left and right terms within the summation end up being the same terms but in reverse orders. Also, since 0 times anything is 0, we can simplify the sum as follows:

$$\boldsymbol{T(n)=n+\frac{1}{n}(1)+\frac{2}{n}\sum_{i=1}^{n-1}\frac{i}{n}T(i)}$$

We can multiply this equation through by n^2:

$$\boldsymbol{n^2T(n)=n^3+n+2\sum_{i=1}^{n-1}iT(i)}$$

Now, plug in n-1 into this equation:

$$\boldsymbol{(n-1)^2T(n-1)=(n-1)^3+(n-1)+2\sum_{i=1}^{n-2}iT(i)}$$

Next, subtract the bottom equation from the top:

$$n^2T(n) - (n-1)^2T(n-1) = (3n^2 - 3n + 2) + 2(n-1)T(n-1)$$

We can combine like terms which are multiples of T(n – 1):

$$n^2T(n) = (n^2 - 2n + 1 + 2n - 2)T(n-1) + (3n^2 - 3n + 2)$$

$$n^2T(n) = (n^2 - 1)T(n-1) + (3n^2 - 3n + 2)$$

$$n^2T(n) = (n-1)(n+1)T(n-1) + (3n^2 - 3n + 2)$$

Now, this step is strange, but divide the whole equation above by n(n+1):

$$\frac{n^2T(n)}{n(n+1)} = \frac{(n-1)(n+1)}{n(n+1)}T(n-1) + \frac{3n^2 - 3n + 2}{n(n+1)}$$

What you can see in this next step is that this particular choice allows products to cancel, leaving a very nice pattern in the recurrence we can exploit.

$$\frac{nT(n)}{(n+1)} = \frac{(n-1)}{n}T(n-1) + \frac{3n^2 - 3n + 2}{n(n+1)}$$

At this point, we can look at the fraction all the way on the left and note that if we were mathematicians, we would simplify that all the way via partial fractions, but really, that fraction a hair under 3 for most reasonable positive integer values of n. This simple substitution (substituting the last fraction for the value 3) will slightly over-estimate the value of T(n), but in such a slight way that it won't affect any order notation answer, (and in fact, the leading constant we'll get will stay accurate):

$$\frac{nT(n)}{(n+1)} = \frac{(n-1)}{n}T(n-1) + 3$$

Finally, let's do the following substitution: Let $S(n) = \frac{nT(n)}{n+1}$. Then we get the following recurrence in S:

$$S(n) = S(n-1) + 3$$

It's easy enough to let S(0) = 0. When we do this, we find that:

$$S(n) = \sum_{i=1}^{n} 3 = 3n$$

Finally, let's solve for T(n):

$$S(n) = \frac{nT(n)}{n+1}$$

$$3n = \frac{nT(n)}{n+1}$$

$$T(n) = 3(n+1)$$

It follows that T(n) = O(n).

Thus, even though the worst case performance of Quick Select is O(n^2), in the average case, we only do about three times amount of work as we would do in the best case.

Note: The real summation we should handle is:

$$S(n) = \sum_{i=1}^{n} \frac{3i^2 - 3i + 2}{i(i+1)}$$

After dividing through by the denominator and using partial fractions, we get:

$$S(n) = \sum_{i=1}^{n} (3 - \left[\frac{8}{i+1} - \frac{2}{i}\right])$$

Once we take care of the telescopic pieces of the sum, we'll get:

$$S(n) = 3n - \left(6(H_n - 1) + \frac{8}{n+1} - 2\right)$$

$$S(n) = 3n + 8 + \frac{8}{n+1} - 6H_n$$

Where, H_n represents the nth Harmonic number.

The corresponding value for T(n) remains extremely close to 3n. As n grows large, the value will be slightly less than 3n. Since we're simply trying to arrive at a Big-Oh bound (because there are many constants not accounted for already), the extra care doesn't lead to a significantly different result than before.