KÅ 2004-09-16

Quick sort analysis

This document gives an alternative way to bind the expected running time of quick sort. It is less strict than Chapter 7.4, but I think it is a good complement.

Consider quick sort of an array with n elements. Let q be the partitioning index returned from the partitioning. The work is then

$$T(n) = T(q-1) + T(n-q) + Dn$$

where Dn represents the work done in the partitioning. However, we do not know beforehand the value of q. We make the assumption that q can be any value between 1 and n with the same probability. Therefore, the work on average is

$$T(n) = \sum_{q=1}^{n} \frac{1}{n} \left(T(q-1) + T(n-q)\right) + Dn.$$

Each $T(k)$ appears twice in the sum.

$$T(n) = \sum_{k=0}^{n-1} \frac{2}{n} T(k) + Dn.$$

We will solve this recursion formula by rewriting it as a "telescoping" summation. First, multiply with n:

$$nT(n) = 2\sum_{k=0}^{n-1} T(k) + Dn^2.$$

Replace n with $n-1$ yields

$$(n-1)T(n-1) = 2\sum_{k=0}^{n-2} T(k) + D(n-1)^2.$$

Subtraction gives

$$nT(n) - (n-1)T(n-1) = 2T(n-1) + 2Dn - D.$$

The last constant term can be neglected, and with some rewriting we obtain

$$nT(n) = (n+1)T(n-1) + 2Dn.$$

Divide with $n(n+1)$.

$$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2D}{n+1}.$$

Again, replace n with $n-1$, and so on until we reach $T(0)$:

$$\begin{aligned}
\frac{T(n)}{n+1} &= \frac{T(n-1)}{n} + \frac{2D}{n+1} \\
\frac{T(n-1)}{n} &= \frac{T(n-2)}{n-1} + \frac{2D}{n} \\
\frac{T(n-2)}{n-1} &= \frac{T(n-3)}{n-2} + \frac{2D}{n-1} \\
&\vdots \\
\frac{T(1)}{2} &= \frac{T(0)}{1} + \frac{2D}{2}
\end{aligned}$$

Now, we get our "telescoping sum" when all these equations are summed. Almost all $T(k)$ vanishes, and we get

$$\frac{T(n)}{n+1} = \frac{T(0)}{1} + 2D \sum_{k=2}^{n+1} \frac{1}{k}. \tag{1}$$

Observe that the summation is similar the harmonic sum H_m,

$$H_m = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$$

The harmonic sum H_m can be bounded by $1 + \int_1^m \frac{1}{x} dx = 1 + \ln m$. As a consequence, $H_m = O(\lg m)$. By using this result in (1) we obtain

$$T(n) = O(n \lg n).$$

The conclusion is, that if q has the same probability to be any of n indices when partitioning the array, then will quick sort have a running time bounded by $O(n \lg n)$.