

Homework 7

Rumen Mitov

March 25, 2024

Problem 7.1

Here was the measured average running time for the three partitioning methods:

Lomoto: 29696 ns

Hoare: 29334 ns

Median of Three: 31876 ns

We can clearly observe that the order (in terms of quickest first) is as follows: Hoare, Lomoto, Median of Three.

Lomoto works by taking the last element of the array as the pivot. This could be inefficient in cases similar to when elements are reverse-sorted already. Median of Three mitigates such cases by taking the median value of the first, last and middle elements of the array as a pivot. However, I presume the computation of the median is what slows down this partitioning scheme overall. Finally, the Hoare method continuously swaps the "big" numbers to the back and the "small" numbers to the front, and unlike the Lomoto method it does this from both ends of the array. In the end, the partitioned array will be "more" sorted than the Lomoto one because in cases where the array is more-or-less reversely sorted, the "smallest" elements will be at the very front, whereas the "largest" will be at the very back, hence garnering fewer swaps in the overall sorting of the array.

Problem 7.1

The best case will be when all three partitions are equal in size. And thus:

$$T(n) = 3T(n/3) + \Theta(n)$$

Using the Master Theorem:

$$a = 3$$

$$b = 3$$

$$\log_b(a) = 1 < f(n) = n$$

$$\implies T(n) = \Omega(n \cdot \log(n)) \text{ (after dropping constants)}$$

The worst case will be when all the pivots are right next to one another at one end of the array, leaving only a single partition of size $n - 2$. Hence:

$$T(n) = T(n - 2) + \Theta(n)$$

For this we can use a Recursive Tree with the height of which will be $\frac{n}{2}$. In each branch we have an operation $\Theta(n)$, hence $T(n) = O(n^2)$ (after dropping the constants).