Quicksort

Michele Chersich, Pratyai Mazumder, Lodovico Mazzei

Quicksort

The sorting problem

Sorting problems

Rearranging a sorted collection of elements

Sorting problems

Rearranging a sorted collection of elements
(usually an n-element array)

Sorting problems

Rearranging a sorted collection of elements
(usually an n-element array)

a_0	a_1	a_2	$\cdots$	a_{n-2}	a_{n-1}	a_n

Input: sequence of n elements

Sorting problems

Rearranging a sorted collection of elements
(usually an n-element array)

a_0	a_1	a_2	$\cdots$	a_{n-2}	a_{n-1}	a_n

Input: sequence of n elements

a'_0	a'_1	a'_2	$\cdots$	a'_{n-2}	a'_{n-1}	a'_n

Output: ordered permutation

Worst-case runtime analysis

$T(n)$ is the number of comparisons, expressed by:

$$T(n) = \max_{1 \leq i \leq n-1} \{T(i) + T(n-i)\} + cn$$
$$T(1) = \Theta(1)$$

where i is the pivot and cn is partitioning time.

The worst choices are $i = 1$ and $i = n - 1$:

$$T(n) = \Theta(n^2)$$

Average runtime analysis

The pivot is equally likely to be any element between 1 and n:

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

$$T(n) = \frac{2}{n} \sum_{i=0}^{n-1} T(i) + cn$$

$$nT(n) = 2 \sum_{i=0}^{n-1} T(i) + cn^2$$

Average runtime analysis

Subtract the previous term of the recurrence relation:

$$nT(n) - (n-1)T(n-1) = 2T(n-1) + 2n - c$$

$$nT(n) - (n+1)T(n-1) = 2cn$$

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

Average runtime analysis

We get the sequence of equations:

$$\begin{aligned}
\frac{T(n)}{n+1} - \frac{T(n-1)}{n} &= \frac{2c}{n+1} \\
\frac{T(n-1)}{n} - \frac{T(n-2)}{n-1} &= \frac{2c}{n} \\
\frac{T(n-2)}{n-1} - \frac{T(n-3)}{n-2} &= \frac{2c}{n-1} \\
&\cdots \\
\frac{T(2)}{3} - \frac{T(1)}{2} &= \frac{2c}{3}
\end{aligned}$$

Average runtime analysis

Adding up all the equations:

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

Considering the Harmonic series:

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

Discarding all constant terms:

$$T(n) = \mathcal{O}(n \log n)$$