K. Mehlhorn e P. Sanders
Algorithms and data structures: the basic Toolbox
Springer, 2008

As for quicksort, the worst-case execution time of quickselect is quadratic. But the expected execution time is linear and hence is a logarithmic factor faster than quicksort.

$|a| = \#$ elementi $<$ perno
↳ $|a| < q$

$|c| = \#$ elementi $>$ perno
↳ $|c| \leq n-q$

Theorem 5.8. *The quickselect algorithm runs in expected time* $O(n)$ *on an input of size* n.

Proof. We shall give an analysis that is simple and shows a linear expected execution time. It does not give the smallest constant possible. Let $T(n)$ denote the expected execution time of quickselect. We call a pivot *good* if neither $|a|$ nor $|c|$ is larger than $2n/3$. Let γ denote the probability that a pivot is good; then $\gamma \geq 1/3$. We now make the conservative assumption that the problem size in the recursive call is reduced only for good pivots and that, even then, it is reduced only by a factor of $2/3$. Since the work outside the recursive call is linear in n, there is an appropriate constant c such that

$$T(n) \leq cn + \gamma T\left(\frac{2n}{3}\right) + (1-\gamma)T(n).$$

• Si assuma che non possano esistere elementi uguali.

Solving for $T(n)$ yields

$$T(n) \leq \frac{cn}{\gamma} + T\left(\frac{2n}{3}\right) \leq 3cn + T\left(\frac{2n}{3}\right) \leq 3c\left(n + \frac{2n}{3} + \frac{4n}{9} + \ldots\right)$$
$$\leq 3cn\sum_{i\geq 0}\left(\frac{2}{3}\right)^i \leq 3cn\frac{1}{1-2/3} = 9cn.$$

□

$$\boxed{\begin{array}{l} |a| = q-1 \\ |c| = n-q \end{array}}$$