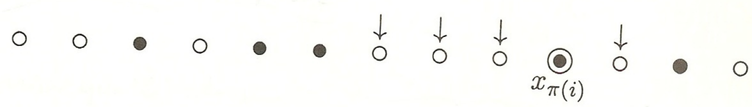
$(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$. Then the expectation of $T(\pi)$ for a random permutation π is at most $2n \ln n$.

Let us remark that *any* algorithm that sorts every n-tuple of distinct real numbers and uses only pairwise comparisons of these numbers has to make at least $\log_2 n!$ comparisons in the worst case. This is because the algorithm must select one of the $n!$ possible permutations according to the outcomes of the comparisons, and k comparisons only have 2^k different outcomes. As we know from Section 2.5, we have $n! \geq \left(\frac{n}{e}\right)^n$, and so $\log_2 n! = (\log_2 e) \ln n! \geq (\log_2 e)(n-1)\ln n \approx 1.443(n-1)\ln n$. Therefore, the average behavior of QUICKSORT guaranteed by Theorem 9.4.4 is quite good.

Reasons for the $O(n \log n)$ average behavior of the QUICKSORT algorithm are not difficult to see. When the input elements are randomly ordered, we expect that the first element usually divides the remaining elements into two groups of roughly comparable size—it is unlikely that one group is much smaller than the other. If this happens in most cases, the recursion in the algorithm will have roughly $\log n$ levels, and at each level of recursion we need $O(n)$ comparisons in total.

This may sound quite convincing, but certainly it is not a proof. Next, we give a rigorous proof, based on a rather different idea.

Proof of Theorem 9.4.4. Let $T_i = T_i(\pi)$ be the number of elements compared to the element $x_{\pi(i)}$ at the moment $x_{\pi(i)}$ is the dividing element. For instance, we always have $T_1 = n - 1$, since $x_{\pi(1)}$ is the first element in the input sequence, and all the other elements are compared to it. If $\pi(2) < \pi(1)$, T_2 is $\pi(1) - 2$, and for $\pi(2) > \pi(1)$ we have $T_2 = n - \pi(1) - 1$. In general, T_i can be interpreted according to the following diagram:



The small circles in this diagram depict the elements $x_1, x_2, \ldots, x_n$ in the sorted order. The full circles are the elements with indices $\pi(1), \pi(2), \ldots, \pi(i-1)$, i.e. the first $i-1$ elements in the input sequence. The element $x_{\pi(i)}$ is marked by a double circle, and the remaining elements have empty circles. It is not difficult to see that T_i is exactly the number of empty circles "seen" by the element $x_{\pi(i)}$, if the full circles are considered opaque.