REMARKS.

1009 2-S-Maintain the following invariants:

(i) \( 1 \leq i \leq r \)

(ii) \( A[p \ldots i] \leq A[r] \)

(iii) \( \| A[i] \| \leq A[(i+1) \ldots (i+1)] \)

(iv) The elements in \( A[i \ldots (r-1)] \) have not been processed and may be \( > A[i] \)
or \( \leq A[i] \).

EXERCISE

- Verify these claims on examples.
- Prove them.

It follows that when Partition returns:

\[
A[p \ldots (q-1)] \leq A[q] \leq A[(q+1) \ldots n]
\]

The run time of Partition (in both worst and average cases) is \( O(n) \) where

\( n = r - p + 1 \), since loop 2-S steps through the entire subarray \( A[p \ldots (r-1)] \),

then the pivot is set in place.
Exercise

Prove the correctness of Quicksort by induction on the length of the sub-array \( A[p..r] \) : \( n = r - p + 1 \).

The run time of Quicksort depends heavily on the value of \( p \) chosen by partition. If the subarrays \( A[p..(q-1)] \) and \( A[q+1..r] \) are not balanced (i.e., of roughly equal size) then performance is inhibited. In this case one recursive call to Quicksort is on a subarray which is inordinately long.

The worst case occurs when the array is already sorted. Then partition returns

\[
A[p...(r-1)] \leq A[q] \leq \ldots \text{empty}
\]

\( \text{sorted} \)

\( \uparrow \)

\( 4 \)

Let \( T(n) \) denote the worst case run time of Quicksort (i.e., with \( A[1..n] \) already sorted.)
Then \( T(n) \) satisfies
\[
T(n) = \begin{cases} 
\Theta(1) & n = 0, 1 \\
T(n-1) + \Theta(n) & n \geq 2
\end{cases}
\]

Simplify this to \( T(n) = T(n-1) + cn \) for definitiveness. By the iteration method
\[
T(n) = cn + T(n-1) \\
= cn + c(n-1) + T(n-2) \\
= cn + c(n-1) + c(n-2) + T(n-3) \\
= c \sum_{i=0}^{k-1} (n-i) + T(n-k) \\
= cnk - \frac{1}{2}c(k-1) + T(n-k).
\]

Choose \( k \) such that \( n-k = 1 \), i.e., \( k = n-1 \).

Then
\[
T(n) = cn(n-1) - \frac{1}{2}c(n-1)(n-2) + \text{const},
\]
\[
= \Theta(n^2).
\]

So what's so quick about quicksort?
The advantage of Quicksort is in its average case.

We assume that all \( n! \) permutations of the input array \( A[1 \ldots n] \) are equally likely, i.e., that any given permutation occurs with probability \( \frac{1}{n!} \).

We choose as basic operation (i.e., benchmark) the comparison of numerical values on line 3 of Partition. Let \( t(n) \) denote the average number of comparisons performed by Quicksort on an input array \( A[1 \ldots n] \) of length \( n \), i.e.,

\[
t(n) = \frac{\sum \text{(all permutations)} \text{(\# of comparisons performed on given permutation)}}{n!}.
\]

We wish to determine a recurrence for \( t(n) \). Our assumption implies that the pivot \( A[\ell] \) is equally likely to be placed in any of the \( n \) locations in \( A[1 \ldots n] \).

Thus, the return value \( \ell \) of Partition has probability \( \frac{1}{n} \) of being any one of the \( n \) values: \( \ell = 1, 2, \ldots, n \).
Observe that Partition itself does \((n-1)\) comparisons on \(A[1\ldots n]\). Also

\[
\text{Length}[A[1\ldots (q-1)]] = q - 1
\]

And

\[
\text{Length}[A[(q+1)\ldots n]] = n - q.
\]

Thus

\[
t(n) = \frac{\sum_{q=1}^{n} \left( (n-1) + \frac{1}{q} \sum_{q=1}^{n} t(q-1) + t(n-q) \right)}{n},
\]

So

\[
t(n) = (n-1) + \frac{1}{n} \sum_{q=1}^{n} \left( t(q-1) + t(n-q) \right).
\]

The initial values \(t(0) = 0, t(1) = 0, t(2) = 1\) can be seen by inspection. Thus

\[
t(n) = (n-1) + \frac{1}{n} \left( \sum_{q=1}^{n-1} t(q) + \sum_{q=1}^{n-1} t(n-q) \right),
\]

whence

\[
t(n) = (n-1) + \frac{2}{n} \sum_{q=1}^{n-1} t(q).
\]
To solve this recurrence, we resort to some tricks. Let

\[ x_n = \sum_{q=1}^{n-1} t(q), \quad x_1 = 0. \]

Then

\[ x_{n+1} - x_n = \sum_{q=1}^{n} t(q) - \sum_{q=1}^{n-1} t(q) = t(n), \]

and so

\[ x_{n+1} - x_n = (n-1) + \frac{2}{n} \cdot x_n. \]

\[ \therefore \quad x_{n+1} - \left(\frac{n+2}{n}\right)x_n = n-1 \]

Multiply by the magic number \(\frac{1}{(n+1)(n+2)}\)

\[ \therefore \quad \frac{x_{n+1}}{(n+1)(n+2)} - \frac{x_n}{n(n+1)} = \frac{n-1}{(n+1)(n+2)} = \frac{3}{n+2} - \frac{2}{n+1} \]

Reduce by \(r\):

\[ \frac{x_{r+1}}{(r+1)(r+2)} - \frac{x_r}{r(r+1)} = \frac{3}{r+2} - \frac{2}{r+1} \]
\[ \sum_{r=1}^{n-1} \left( \frac{x_{r+1}}{(r+1)(r+2)} - \frac{x_r}{r(r+1)} \right) = \sum_{r=1}^{n-1} \left( \frac{1}{r+2} + \frac{2}{r^2} - \frac{2}{r+1} \right) \]

\[ \sum_{r=2}^{n} \frac{x_r}{\sqrt{r+1}} - \sum_{r=1}^{n-1} \frac{x_r}{\sqrt{r+1}} = \sum_{r=3}^{n+1} \frac{1}{r} + \sum_{r=3}^{n+1} \frac{2}{r} - \sum_{r=2}^{n} \frac{2}{r} \]

\[ \frac{x_n}{n(n+1)} - \frac{x_1}{2} = \sum_{r=3}^{n+1} \frac{1}{r} + \frac{2}{n+1} - 1 \]

\[ = \sum_{r=1}^{n} \frac{1}{r} + \frac{1}{n+1} - 1 - \frac{1}{2} + \frac{2}{n+1} - 1 \]

\[ = \sum_{r=1}^{n} \frac{1}{r} + \frac{3}{n+1} - \frac{5}{2} \]

Recall \( x_1 = 0 \). Define \( H_n = \sum_{r=1}^{n} \frac{1}{r} \) (called the \( n \text{th} \) harmonic number). Then

\[ \frac{x_n}{n(n+1)} = \frac{3}{n+1} - \frac{5}{2} + H_n \]

\[ \therefore x_n = 3n - \frac{5}{2} n(n+1) + n(n+1)H_n \]

\[ \therefore t(n) = (n-1) + \frac{2}{n} x_n \]
\[ t(n) = -4n + 2(n+1) \cdot H_n \]

We must estimate the size of \( H_n \) to determine the asymptotic order of \( t(n) \).

\[ \text{Area } A = H_n = \frac{1}{2} \sum_{r=1}^{n} \frac{1}{r} \]

\[ \int_{1}^{n+1} \frac{1}{u} \, du \leq H_n \leq 1 + \int_{1}^{n} \frac{1}{u} \, du \]