QuickSort Partition Code — Annotated
Jonathan Buss, Collin Roberts

Algorithm and Code

Overview of the Code

Array to partition: $X$, indexed from 1 to $n$.

Pivot: $p$. It may have any value.

First while-loop:

Positions the $a$ marker to the first element which is greater than $p$. If no such element exists, $a$ becomes $n$, and the second while-loop will skip its body.

Second while-loop:

Move the $b$ marker over any remaining elements, and swap an out-of-position one into place. After a swap, we move the $a$ marker one position to the right (a large element moved out and a small one into its former place).

Postcondition:

$z$ is the cutoff position between the “small” elements on the left and the “large” elements on the right.

Code (unannotated)

$$\begin{align*}
    & (n \geq 1) \\
    & a = 1 \\
    & \text{while} \ (a < n \land X[a] \leq p) \ { \{ \} } \\
    & \quad a = a + 1 \\
    & b = a + 1 \\
    & \text{while} \ (b \leq n) \ { \{ \\
    & \quad \text{if} \ (X[b] \leq p) \ { \{ \\
    & \quad \quad t = X[b] ; X[b] = X[a] ; X[a] = t \\
    & \quad \quad a = a + 1 \\
    & \quad \} \\
    & \quad b = b + 1 \\
    & \} \\
    & \ (\exists z \ (1 \leq z \leq n + 1 \land X[1..z] \leq p \land (X[z..n] > p)))
\end{align*}$$

We shall annotate this code in two parts—one for each of the while-loops. To complete the annotation then requires additional implied conditions.
Annotations

Notes on Notation:

- “X[a..b]” refers to all X[i], where a ≤ i < b. Similarly, X[a..b] refers to all X[i], where a ≤ i ≤ b. (This is analogous to the notation for open and closed intervals in calculus.)

- The various parts of formulas are coloured to show their origin and role.
  Greenish: the “lower” (or only) part of a loop invariant.
  Bluish: the “upper” part of a loop invariant, if any.
  Reddish: Conditions from loop guards and conditionals.

Before an assignment, the assigned value has its colour reddened.

Preamble and first loop (invariant 1 ≤ a ∧ X[1..a] ≤ p)

\[
\begin{align*}
(1 \leq 1 \leq n) \land X[1..1] \leq p \\
& \text{(precondition)} \\
a = 1 ; \\
& \text{assignment} \\
\text{while} \ (a < n \land \text{X[a] <= p}) \{ \\
& (1 \leq a \leq n) \land X[1..a] \leq p \land (a < n \land X[a] \leq p) \\
& \text{partial-while implied} \\
a = a + 1 ; \\
& \text{assignment} \\
\}
\end{align*}
\]

Remarks:

- The implication involves a notational shift: X[1..a + 1] ≤ p is equivalent to X[1..a] ≤ p ∧ X[a] ≤ p.
Second loop (invariant \(1 \leq a \leq n\) \& \(X[1..a] \leq p \land X[a..\min(b, n)) > p\))

\[
\begin{align*}
\{X[1..a] \leq p \land X[a..\min(a + 1, n)) > p\} & \quad \text{precondition} \\
& \text{assignment} \\
b = a + 1 ; \\
\{X[1..a] \leq p \land X[a..\min(b, n)) > p\} & \quad \text{partial-while} \\
\text{while } (b \leq n) \{ & \\
\{X[1..a] \leq p \land X[a..b] > p \land b \leq n\} & \quad \text{implied} \\
\{X[1..a] \leq p \land X[a..b] > p \land X[a..b] \leq p\} & \quad \text{if-then} \\
t = X[b] ; X[b] = X[a] ; X[a] = t ; & \\
\{X[1..a] \leq p \land X[a..b] \leq p \land X[a..b] > p\} & \quad \text{swap} \\
\{X[1..a + 1] \leq p \land X[a..b + 1] > p\} & \quad \text{implied} \\
a = a + 1 ; & \\
\{X[1..a] \leq p \land X[a..b + 1] > p\} & \quad \text{assignment} \\
\} & \quad \text{if-then + implied} \\
b = b + 1 \\
\{X[1..a] \leq p \land X[a..b] > p\} & \quad \text{assignment} \\
\{X[1..a] \leq p \land X[a..\min(b, n)) > p\} & \quad \text{implied} \\
\} & \quad \text{partial-while} \\
\{X[1..a] \leq p \land X[a..\min(b, n)) > p \land b > n\} & \quad \text{partial-while} \\
\end{align*}
\]

Remarks:

- The presence of “\(\min(b, n)\)” in the loop invariant accounts for the possibility that \(a\) advanced to \(n\) in the first loop. The relationship of \(X[n]\) to \(p\) is unknown and immaterial. (We could instead include an additional test in the code, but such a test is not needed for correctness.)

  The adjustment is not needed, and hence disappears, inside of the loop:
  
  - At the start of the while-loop, we have \(b \leq n\), and the minimum is always \(b\).
  - At the end of the loop, \(X[a..b] > p\) implies \(X[\min(a, b), n] > p\), whatever value \(b\) has.

- The condition labelled “swap” follows from the proof in the course notes: the three assignments exchange the values of \(X[a]\) and \(X[b]\), leaving the rest of the array unchanged.

The implieds before and after the swap simply re-write the notation, to focus on the changing locations.

Inter-loop implications

After first while-loop:

\[
\begin{align*}
\{X[1..a] \leq p \land (a \geq n \lor X[a] > p)\} & \quad \text{partial-while} \\
\{X[1..a] \leq p \land X[a..\min(a + 1, n)) > p\} & \quad \text{implied} \\
\{X[1..a] \leq p \land X[a..\min(a + 1, n)) > p\} & \quad \text{pre-condition for assignment to \(\exists\)} \\
\end{align*}
\]

Algebra and use of notation.

At end:

\[
\begin{align*}
\{X[1..a] \leq p \land X[\min(b, n)) > p \land b > n\} & \quad \text{partial-while} \\
(\exists z \ (1 \leq z \leq n + 1 \land X[1..z] \leq p \land (X[z..n] > p)) & \quad \text{postcondition} \\
\end{align*}
\]

The value for \(z\) is \(a\) if \(a \leq n\) and \(X[n] > p\), or \(n + 1\) otherwise.
Full annotated code

\{ n \geq 1 \}
\{ X[1..1] \leq p \}\quad \text{implied}
\begin{align*}
a &= 1 ; \\
&\{ X[1..a] \leq p \}\quad \text{assignment}
\end{align*}
while (a < n & X[a] \leq p ) {
\begin{align*}
&\{ X[1..a] \leq p \land (a < n \land X[a] \leq p) \}\quad \text{partial-while} \\
&\{ X[1..a+1] \leq p \}\quad \text{implied}
\end{align*}
a = a + 1 ; \\
\{ X[1..a] \leq p \}\quad \text{assignment}
}
\{ X[1..a] \leq p \land (a \geq n \lor X[a] > p) \}\quad \text{partial-while}
\{ X[1..a] \leq p \land X[a..\min(a+1, n)] > p \}\quad \text{implied}
b = a + 1 ; \\
\{ X[1..a] \leq p \land X[a..\min(b, n)] > p \}\quad \text{assignment}
while (b <= n ) {
\begin{align*}
&(\{ X[1..a] \leq p \land X[a..\min(b, n)] > p \} \land b \leq n) \quad \text{partial-while} \\
&\{ X[1..a] \leq p \land X[a..b] > p \land b \leq n \}\quad \text{implied}
\end{align*}
if (X[b] \leq p ) {
\begin{align*}
&(\{ X[1..a] \leq p \land X[a..b] > p \land b \leq n \land X[b] \leq p \} \quad \text{if-then} \\
&(\{ X[1..a] \leq p \land X[a..b] > p \land X[a+1..b] > p \land X[b] \leq p \}) \quad \text{implied}
\end{align*}
t = X[b] ; X[b] = X[a] ; X[a] = t ; \\
\begin{align*}
&(\{ X[1..a] \leq p \land X[a..b] \leq p \land X[a+1..b] > p \land X[b] > p \}) \quad \text{swap} \\
&(\{ X[1..a+1] \leq p \land X[a+1..b+1] > p \}) \quad \text{implied}
\end{align*}
a = a + 1 ; \\
\{ X[1..a] \leq p \land X[a..b+1] > p \}\quad \text{assignment}
}
\{ X[1..a] \leq p \land X[a..b+1] > p \}\quad \text{if-then} + \text{implied}
b = b + 1 \\
\{ X[1..a] \leq p \land X[a..b] > p \}\quad \text{assignment}
\{ X[1..a] \leq p \land X[a..\min(b, n)] > p \}\quad \text{implied}
}
\{ X[1..a] \leq p \land X[a..\min(b, n)] > p \land b > n \}\quad \text{partial-while}
\{ \exists z (1 \leq z \leq n+1 \land X[1..z] \leq p \land (X[z..n] > p)) \}\quad \text{implied}

Remarks:
- The proof of termination (i.e. the last ingredient required for total correctness) is left as an exercise.