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)

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

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 \((X[1..a) ≤ p])\)

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

Remarks:

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

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..a + 1] > p)\} \quad \text{(precondition)}
\]

\[
b = a + 1 ;
\]

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p)\} \land b \leq n \quad \text{assignment}
\]

\[
\text{while (} b \leq n \text{) }
\]

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p)\} \land b \leq n \quad \text{partial-while}
\]

\[
\left\{ \begin{array}{l}
\{(X[1..a] \leq p) \land X[a..b] > p \land b \leq n \land X[b] \leq p\} \\
\{(X[1..a] \leq p) \land X[a] > p \land X[a + 1..b] > p \land X[b] \leq p\}
\end{array} \right. \quad \text{if-then}
\]

\[
t = X[b] ; X[b] = X[a] ; X[a] = t ;
\]

\[
\{(X[1..a] \leq p) \land X[a] \leq p \land X[a + 1..b] > p \land X[b] > p\} \quad \text{swap}
\]

\[
\{(X[1..a + 1] \leq p) \land X[a + 1..b + 1] > p\} \quad \text{implied}
\]

\[
a = a + 1 ;
\]

\[
\{(X[1..a] \leq p) \land X[a..b + 1] > p\} \quad \text{assignment}
\]

\[
\{(X[1..a] \leq p) \land X[a..b + 1] > p\} \quad \text{if-then + implied}
\]

\[
b = b + 1
\]

\[
\{(X[1..a] \leq p) \land X[a..b] > p\} \quad \text{assignment}
\]

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p)\} \quad \text{implied}
\]

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p) \land b > n\} \quad \text{partial-while}
\]

Remarks:

- The presence of "\((a = n) \lor\)" in the loop invariant accounts for the possibility that \(a\) advanced to \(n\) in the first loop. In this case, 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 disappears inside of the loop; the loop guard \(b \leq n\) makes it irrelevant.

- 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:

\[
\{(X[1..a] \leq p) \land (a \geq n \lor X[a] > p)\} \quad \text{partial-while}
\]

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..a + 1] > p)\} \quad \text{implied}
\]

(pre-condition for assignment to \(a\))

Algebra and use of notation.

At end:

\[
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p) \land b > n\} \quad \text{partial-while}
\]

\[
\exists z (1 \leq z \leq n + 1 \land X[1..z] \leq p \land (X[z..n] > p)) \quad \text{implied (postcondition)}
\]

The "implied" requires a three-way case analysis:

- If \(a < n\), take \(a\) for \(z\).
- If \(a = n \land X[n] \leq p\), take \(n\) for \(z\).
- If \(a = n \land X[n] > p\), take \(n + 1\) for \(z\).
Full annotated code

\{n \geq 1\}
\{(X[1..1] \leq p)\}  \text{ implied}

\text{a} = 1 ;
\{(X[1..a] \leq p)\}  \text{ assignment}

\text{while} ( \text{ a } < n \text{ } \&\& \text{ X[a] } \leq p ) \{ \\
\{ (X[1..a] \leq p) \land (a < n \land X[a] \leq p) \}\}  \text{ partial-while}
\{(X[1..a+1] \leq p)\}  \text{ implied}

\text{a} = \text{a} + 1 ;
\{(X[1..a] \leq p)\}  \text{ assignment}

\} \text{ partial-while}
\{(X[1..a] \leq p) \land (a \geq n \lor X[a] > p)\}
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..a+1] > p)\}  \text{ implied}

\text{b} = \text{a} + 1 ;
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p)\}  \text{ assignment}

\text{while} ( \text{ b } \leq n ) \{ \\
\{ (X[1..a] \leq p) \land (a = n) \lor X[a..b] > p \land b \leq n \}\}  \text{ partial-while}
\{(X[1..a+1] \leq p) \land X[a..b] > p \land b \leq n \}\}  \text{ implied}

\text{if} ( X[b] \leq p ) \{ \\
(X[1..a] \leq p) \land X[a..b] > p \land b \leq n \land X[b] \leq p\}
\{(X[1..a] \leq p) \land X[a] > p \land X[a..b] > p \land X[b] \leq p\}  \text{ implied}

\text{t} = X[b] ; \text{ X[b] = X[a] ; X[a] = t ;}
\{(X[1..a] \leq p) \land X[a] \leq p \land X[a..b] > p \land X[b] > p\}  \text{ swap}
\{(X[1..a+1] \leq p) \land X[a+1..b+1] > p\}  \text{ implied}

\text{a} = \text{a} + 1 ;
\{(X[1..a] \leq p) \land X[a..b+1] > p\}  \text{ assignment}

\} \text{ if-then + implied}
\{(X[1..a] \leq p) \land X[a..b+1] > p\}
\text{b} = \text{b} + 1
\{(X[1..a] \leq p) \land X[a..b] > p\}  \text{ assignment}
\{(X[1..a] \leq p) \land (a = n) \lor X[a..b] > p\}  \text{ implied}

\} \text{ partial-while}
\{(X[1..a] \leq p) \land ((a = n) \lor X[a..b] > p) \land b > n\}  \text{ implied}
\{(\exists z \; (1 \leq z \leq n + 1 \land X[1..z] \leq p \land (X[z..n] > p))\}

Remarks:

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