MODULE *QuickSort*

EXTENDS *Sequences*, *Integers*, *TLC*

CONSTANT *listLength*
ASSUME *listLength* $\in$ *Nat*

```
--algorithm quicksort{
  variables
      indices = 0 .. listLength,
      values = indices,
      listToSort ∈ [indices → values],
      partitionIndex = -1;

  procedure partition( low = 0, high = 0 )
  variable
      pivot = listToSort[high],
      i = (low - 1);
      j = low;
      swapTemp = -1;
  {
      while ( j < high ) {
          if ( listToSort[j] ≤ pivot ) {
                i := i + 1;
                swapTemp := listToSort[i];
                listToSort[i] := listToSort[j];
                listToSort[j] := swapTemp;
           } ;
          j := j + 1;
       } ;

      swapTemp := listToSort[i + 1];
      listToSort[i + 1] := listToSort[high];
      listToSort[high] := swapTemp;

      partitionIndex := i + 1;
      return;
   }

  procedure quickSort( low = 0, high = 0 )
  variable pivot = -1;
  {
      if ( low < high ) {
            call partition(low, high);
            call quickSort(low, partitionIndex - 1);
            call quickSort(partitionIndex + 1, high);
       } ;
      return;
```

```
    }

    {
        call quickSort(0, listLength) ;
        assert ∀ x ∈ 0 .. (listLength − 1) : listToSort[x] ≤ listToSort[x + 1] ;
    }
}
```

BEGIN TRANSLATION ($chksum(pcal) =$ "9118f5ba" $\wedge\ chksum(tla) =$ "2f45a242")
Procedure variable pivot of procedure partition at line 16 col 9 changed to *pivot_*
Parameter low of procedure partition at line 14 col 25 changed to *low_*
Parameter high of procedure partition at line 14 col 33 changed to *high_*

VARIABLES *indices*, *values*, *listToSort*, *partitionIndex*, *pc*, *stack*, *low_*, *high_*, *pivot_*, *i*, *j*, *swapTemp*, *low*, *high*, *pivot*

$vars \triangleq \langle indices, values, listToSort, partitionIndex, pc, stack, low\_, high\_, pivot\_, i, j, swapTemp, low, high, pivot \rangle$

$Init \triangleq$ Global variables
$\wedge\ indices = 0 \,..\, listLength$
$\wedge\ values = indices$
$\wedge\ listToSort \in [indices \to values]$
$\wedge\ partitionIndex = -1$
Procedure partition
$\wedge\ low\_ = 0$
$\wedge\ high\_ = 0$
$\wedge\ pivot\_ = listToSort[high\_]$
$\wedge\ i = (low\_ - 1)$
$\wedge\ j = low\_$
$\wedge\ swapTemp = -1$
Procedure *quickSort*
$\wedge\ low = 0$
$\wedge\ high = 0$
$\wedge\ pivot = -1$
$\wedge\ stack = \langle\rangle$
$\wedge\ pc =$ "Lbl_9"

$Lbl\_1 \triangleq$ $\wedge\ pc =$ "Lbl_1"
$\wedge$ IF $j < high\_$
THEN $\wedge$ IF $listToSort[j] \leq pivot\_$
THEN $\wedge\ i' = i + 1$
$\wedge\ swapTemp' = listToSort[i']$
$\wedge\ listToSort' = [listToSort$ EXCEPT $![i'] = listToSort[j]]$
$\wedge\ pc' =$ "Lbl_2"
ELSE $\wedge\ pc' =$ "Lbl_3"
$\wedge$ UNCHANGED $\langle listToSort, i, swapTemp \rangle$

$$
\begin{array}{l}
\quad \text{ELSE} \quad \land\ swapTemp' = listToSort[i+1] \\
\qquad\qquad \land\ listToSort' = [listToSort \ \text{EXCEPT}\ ![i+1] = listToSort[high\_]] \\
\qquad\qquad \land\ pc' = \text{“Lbl\_4”} \\
\qquad\qquad \land\ i' = i \\
\land\ \text{UNCHANGED}\ \langle indices,\ values,\ partitionIndex,\ stack,\ low\_,\ high\_, \\
\qquad\qquad\qquad pivot\_,\ j,\ low,\ high,\ pivot\rangle
\end{array}
$$

$$
\begin{array}{ll}
Lbl\_3 \triangleq & \land\ pc = \text{“Lbl\_3”} \\
& \land\ j' = j + 1 \\
& \land\ pc' = \text{“Lbl\_1”} \\
& \land\ \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ stack, \\
& \qquad\qquad low\_,\ high\_,\ pivot\_,\ i,\ swapTemp,\ low,\ high,\ pivot\rangle
\end{array}
$$

$$
\begin{array}{ll}
Lbl\_2 \triangleq & \land\ pc = \text{“Lbl\_2”} \\
& \land\ listToSort' = [listToSort \ \text{EXCEPT}\ ![j] = swapTemp] \\
& \land\ pc' = \text{“Lbl\_3”} \\
& \land\ \text{UNCHANGED}\ \langle indices,\ values,\ partitionIndex,\ stack,\ low\_,\ high\_, \\
& \qquad\qquad pivot\_,\ i,\ j,\ swapTemp,\ low,\ high,\ pivot\rangle
\end{array}
$$

$$
\begin{array}{ll}
Lbl\_4 \triangleq & \land\ pc = \text{“Lbl\_4”} \\
& \land\ listToSort' = [listToSort \ \text{EXCEPT}\ ![high\_] = swapTemp] \\
& \land\ partitionIndex' = i + 1 \\
& \land\ pc' = Head(stack).pc \\
& \land\ pivot\_' = Head(stack).pivot\_ \\
& \land\ i' = Head(stack).i \\
& \land\ j' = Head(stack).j \\
& \land\ swapTemp' = Head(stack).swapTemp \\
& \land\ low\_' = Head(stack).low\_ \\
& \land\ high\_' = Head(stack).high\_ \\
& \land\ stack' = Tail(stack) \\
& \land\ \text{UNCHANGED}\ \langle indices,\ values,\ low,\ high,\ pivot\rangle
\end{array}
$$

$$
partition \triangleq Lbl\_1 \lor Lbl\_3 \lor Lbl\_2 \lor Lbl\_4
$$

$$
\begin{array}{ll}
Lbl\_5 \triangleq & \land\ pc = \text{“Lbl\_5”} \\
& \land\ \text{IF}\ low < high \\
& \quad \text{THEN}\ \land\ \land\ high\_' = high \\
& \qquad\qquad\qquad \land\ low\_' = low \\
& \qquad\qquad\qquad \land\ stack' = \langle [procedure \ \mapsto\ \text{“partition”}, \\
& \qquad\qquad\qquad\qquad\qquad pc \ \mapsto\ \text{“Lbl\_6”}, \\
& \qquad\qquad\qquad\qquad\qquad pivot\_ \ \mapsto\ pivot\_, \\
& \qquad\qquad\qquad\qquad\qquad i \ \mapsto\ i, \\
& \qquad\qquad\qquad\qquad\qquad j \ \mapsto\ j, \\
& \qquad\qquad\qquad\qquad\qquad swapTemp \ \mapsto\ swapTemp, \\
& \qquad\qquad\qquad\qquad\qquad low\_ \ \mapsto\ low\_, \\
& \qquad\qquad\qquad\qquad\qquad high\_ \ \mapsto\ high\_]\rangle
\end{array}
$$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad \circ\ stack \\
\qquad\qquad \wedge\ pivot\_' = listToSort[high\_'] \\
\qquad\qquad \wedge\ i' = (low\_' - 1) \\
\qquad\qquad \wedge\ j' = low\_' \\
\qquad\qquad \wedge\ swapTemp' = -1 \\
\qquad\qquad \wedge\ pc' = \text{“Lbl\_1”} \\
\qquad \text{ELSE} \quad \wedge\ pc' = \text{“Lbl\_8”} \\
\qquad\qquad \wedge\ \text{UNCHANGED}\ \langle stack,\ low\_,\ high\_,\ pivot\_,\ i,\ j,\ swapTemp\rangle \\
\wedge\ \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ low, \\
\qquad\qquad high,\ pivot\rangle
\end{array}
$$

$$
\begin{array}{lll}
Lbl\_6 \triangleq & \wedge\ pc = \text{“Lbl\_6”} \\
& \wedge\ \wedge\ high' = partitionIndex - 1 \\
& \quad \wedge\ low' = low \\
& \quad \wedge\ stack' = \langle [procedure \mapsto \text{“quickSort”}, \\
& \qquad\qquad\qquad pc \mapsto \text{“Lbl\_7”}, \\
& \qquad\qquad\qquad pivot \mapsto pivot, \\
& \qquad\qquad\qquad low \mapsto low, \\
& \qquad\qquad\qquad high \mapsto high] \rangle \\
& \qquad\qquad\qquad \circ\ stack \\
& \wedge\ pivot' = -1 \\
& \wedge\ pc' = \text{“Lbl\_5”} \\
& \wedge\ \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ low\_, \\
& \qquad\qquad high\_,\ pivot\_,\ i,\ j,\ swapTemp\rangle
\end{array}
$$

$$
\begin{array}{lll}
Lbl\_7 \triangleq & \wedge\ pc = \text{“Lbl\_7”} \\
& \wedge\ \wedge\ high' = high \\
& \quad \wedge\ low' = partitionIndex + 1 \\
& \quad \wedge\ stack' = \langle [procedure \mapsto \text{“quickSort”}, \\
& \qquad\qquad\qquad pc \mapsto \text{“Lbl\_8”}, \\
& \qquad\qquad\qquad pivot \mapsto pivot, \\
& \qquad\qquad\qquad low \mapsto low, \\
& \qquad\qquad\qquad high \mapsto high] \rangle \\
& \qquad\qquad\qquad \circ\ stack \\
& \wedge\ pivot' = -1 \\
& \wedge\ pc' = \text{“Lbl\_5”} \\
& \wedge\ \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ low\_, \\
& \qquad\qquad high\_,\ pivot\_,\ i,\ j,\ swapTemp\rangle
\end{array}
$$

$$
\begin{array}{lll}
Lbl\_8 \triangleq & \wedge\ pc = \text{“Lbl\_8”} \\
& \wedge\ pc' = Head(stack).pc \\
& \wedge\ pivot' = Head(stack).pivot \\
& \wedge\ low' = Head(stack).low \\
& \wedge\ high' = Head(stack).high \\
& \wedge\ stack' = Tail(stack) \\
& \wedge\ \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ low\_,
\end{array}
$$

$$high\_,\ pivot\_,\ i,\ j,\ swapTemp\rangle$$

$quickSort \triangleq Lbl\_5 \lor Lbl\_6 \lor Lbl\_7 \lor Lbl\_8$

$$\begin{aligned}
Lbl_9 \triangleq\ & \land pc = \text{“Lbl_9”} \\
& \land\ \land high' = listLength \\
& \quad \land low' = 0 \\
& \quad \land stack' = \langle [procedure \mapsto \text{“quickSort”}, \\
& \qquad\qquad pc \mapsto \text{“Lbl_10”}, \\
& \qquad\qquad pivot \mapsto pivot, \\
& \qquad\qquad low \mapsto low, \\
& \qquad\qquad high \mapsto high] \rangle \\
& \qquad\qquad \circ stack \\
& \land pivot' = -1 \\
& \land pc' = \text{“Lbl_5”} \\
& \land \text{UNCHANGED } \langle indices,\ values,\ listToSort,\ partitionIndex,\ low_, \\
& \qquad\qquad high_,\ pivot_,\ i,\ j,\ swapTemp \rangle
\end{aligned}$$

$$\begin{aligned}
Lbl_10 \triangleq\ & \land pc = \text{“Lbl_10”} \\
& \land Assert(\forall\, x \in 0\,..\,(listLength - 1) : listToSort[x] \leq listToSort[x+1], \\
& \qquad\qquad \text{“Failure of assertion at line 53, column 9.”}) \\
& \land pc' = \text{“Done”} \\
& \land \text{UNCHANGED } \langle indices,\ values,\ listToSort,\ partitionIndex,\ stack, \\
& \qquad\qquad low_,\ high_,\ pivot_,\ i,\ j,\ swapTemp,\ low,\ high, \\
& \qquad\qquad pivot \rangle
\end{aligned}$$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq pc = \text{“Done”} \land \text{UNCHANGED } vars$

$$\begin{aligned}
Next \triangleq\ & partition \lor quickSort \lor Lbl_9 \lor Lbl_10 \\
& \lor Terminating
\end{aligned}$$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION

* Modification History
* Last modified *Tue Mar* 12 21:16:15 *CET* 2024 by *jeujeus*
* Created *Tue Mar* 12 18:38:34 *CET* 2024 by *jeujeus*