MODULE *QuickSort*

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

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

```
--algorithm quicksort{
  variables
      $indices = 0 \mathrel{..} listLength$,
      $values = indices$,
      $listToSort \in [indices \rightarrow values]$,
      $partitionIndex = -1$ ;

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

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

      $partitionIndex := i + 1$ ;
      return ;
  }

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

```
    {
        call $quickSort(0, listLength)$ ;
        assert $\forall x \in 0 \,..\, (listLength - 1) : listToSort[x] \leq listToSort[x + 1]$ ;
     }
}
```

BEGIN TRANSLATION ($chksum(pcal) =$ "62fb6ce4" $\land\ chksum(tla) =$ "15f114fc")

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

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

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

$Lbl\_1 \triangleq$ $\land\ pc =$ "Lbl_1"
         $\land$ IF $j < high\_$
             THEN $\land$ IF $listToSort[j] \leq pivot$
                          THEN $\land\ i' = i + 1$
                               $\land\ swapTemp' = listToSort[i']$
                               $\land\ listToSort' = [listToSort$ EXCEPT $![i'] = listToSort[j]]$
                               $\land\ pc' =$ "Lbl_2"
                          ELSE $\land\ pc' =$ "Lbl_3"
                               $\land$ UNCHANGED $\langle listToSort, i, swapTemp \rangle$
             ELSE $\land\ swapTemp' = listToSort[i + 1]$
                  $\land\ listToSort' = [listToSort$ EXCEPT $![i + 1] = listToSort[high\_]]$
                  $\land\ pc' =$ "Lbl_4"
```

$$\begin{array}{l}
\qquad\qquad \land\ i' = i \\
\land\ \textsc{unchanged}\ \langle \mathit{indices},\ \mathit{values},\ \mathit{partitionIndex},\ \mathit{stack},\ \mathit{low\_},\ \mathit{high\_}, \\
\qquad\qquad\qquad \mathit{pivot},\ j,\ \mathit{low},\ \mathit{high} \rangle
\end{array}$$

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

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

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

$$\mathit{partition} \triangleq \mathit{Lbl\_1} \lor \mathit{Lbl\_3} \lor \mathit{Lbl\_2} \lor \mathit{Lbl\_4}$$

$$\begin{array}{ll}
\mathit{Lbl\_5} \triangleq & \land\ \mathit{pc} = \text{“Lbl\_5”} \\
& \land\ \textsc{if}\ \mathit{low} < \mathit{high} \\
& \qquad \textsc{then}\ \land\ \land\ \mathit{high\_}' = \mathit{high} \\
& \qquad\qquad\qquad \land\ \mathit{low\_}' = \mathit{low} \\
& \qquad\qquad\qquad \land\ \mathit{stack}' = \langle [\begin{array}{lcl}
\mathit{procedure} & \mapsto & \text{“partition”}, \\
\mathit{pc} & \mapsto & \text{“Lbl\_6”}, \\
\mathit{pivot} & \mapsto & \mathit{pivot}, \\
i & \mapsto & i, \\
j & \mapsto & j, \\
\mathit{swapTemp} & \mapsto & \mathit{swapTemp}, \\
\mathit{low\_} & \mapsto & \mathit{low\_}, \\
\mathit{high\_} & \mapsto & \mathit{high\_}] \rangle
\end{array} \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \circ\ \mathit{stack} \\
& \qquad\qquad\quad \land\ \mathit{pivot}' = \mathit{listToSort}[\mathit{high\_}'] \\
& \qquad\qquad\quad \land\ i' = (\mathit{low\_}' - 1)
\end{array}$$

```
                  $\land j' = low\_'$
                  $\land swapTemp' = -1$
                  $\land pc' = \text{“Lbl\_1”}$
            ELSE  $\land pc' = \text{“Lbl\_8”}$
                  $\land$ UNCHANGED $\langle stack, low\_, high\_, pivot, i, j, swapTemp\rangle$
        $\land$ UNCHANGED $\langle indices, values, listToSort, partitionIndex, low,$
                          $high\rangle$

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

$Lbl\_7 \triangleq$ $\land pc = \text{“Lbl\_7”}$
        $\land$ $\land high' = high$
          $\land low' = partitionIndex + 1$
          $\land stack' = \langle[procedure \mapsto \text{“quickSort”},$
                             $pc \mapsto \text{“Lbl\_8”},$
                             $low \mapsto low,$
                             $high \mapsto high]\rangle$
                             $\circ\ stack$
        $\land pc' = \text{“Lbl\_5”}$
        $\land$ UNCHANGED $\langle indices, values, listToSort, partitionIndex, low\_,$
                          $high\_, pivot, i, j, swapTemp\rangle$

$Lbl\_8 \triangleq$ $\land pc = \text{“Lbl\_8”}$
        $\land pc' = Head(stack).pc$
        $\land low' = Head(stack).low$
        $\land high' = Head(stack).high$
        $\land stack' = Tail(stack)$
        $\land$ UNCHANGED $\langle indices, values, listToSort, partitionIndex, low\_,$
                          $high\_, pivot, i, j, swapTemp\rangle$

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

$Lbl\_9 \triangleq$ $\land pc = \text{“Lbl\_9”}$
        $\land$ $\land high' = listLength$
          $\land low' = 0$
          $\land stack' = \langle[procedure \mapsto \text{“quickSort”},$
```

$$\begin{array}{l}
\qquad\qquad\qquad\qquad pc \;\mapsto\; \text{“Lbl\_10”}, \\
\qquad\qquad\qquad\qquad low \;\mapsto\; low, \\
\qquad\qquad\qquad\qquad high \;\mapsto\; high]\rangle \\
\qquad\qquad\qquad\qquad \circ\, stack \\
\qquad\quad \land\; pc' = \text{“Lbl\_5”} \\
\qquad\quad \land\; \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ low\_, \\
\qquad\qquad\qquad\quad high\_,\ pivot,\ i,\ j,\ swapTemp\rangle
\end{array}$$

$$\begin{array}{lll}
Lbl\_10 & \triangleq & \land\; pc = \text{“Lbl\_10”} \\
& & \land\; Assert(\forall\, x \in 0 \,..\, (listLength - 1) : listToSort[x] \leq listToSort[x+1], \\
& & \qquad\qquad \text{“Failure of assertion at line 52, column 9.”}) \\
& & \land\; pc' = \text{“Done”} \\
& & \land\; \text{UNCHANGED}\ \langle indices,\ values,\ listToSort,\ partitionIndex,\ stack, \\
& & \qquad\qquad\qquad low\_,\ high\_,\ pivot,\ i,\ j,\ swapTemp,\ low,\ high\rangle
\end{array}$$

Allow infinite stuttering to prevent deadlock on termination.

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

$$\begin{array}{lll}
Next & \triangleq & partition \lor quickSort \lor Lbl\_9 \lor Lbl\_10 \\
& & \quad \lor\; Terminating
\end{array}$$

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

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

END TRANSLATION

* Modification History

* Last modified *Wed Mar* 13 09:00:34 *CET* 2024 by *jeujeus*

* Created *Tue Mar* 12 18:38:34 *CET* 2024 by *jeujeus*