

AP COMPUTER SCIENCE A – QUICK SORT – NOTES

The program `QuickSortInt.java` sorts a list of integers into ascending order using the so-called quick sort algorithm. The method `public long quickSort ()` (on lines 35 through 47) starts the sort with a call to `partition (lo, hi)` at line 43, *i.e.*, the sort starts with the partition of entries running from `lo` index 0 to `hi` index `len-1`, inclusive (which is the entire array). The method `private void partition (int lo, int hi)` (on lines 81 through 108) is the method which actually performs the sort. Note that `partition` calls itself (on lines 104 and 105) so that the algorithm is recursive.

Below is the state of an array of 10 entries after each pass (at line 102) of the sort. Note that the leftmost column contains the original (unsorted) list.

| o r i g i n a l | pass 1 | pass 2 | pass 3 | pass 4 |
|--------------------------------------|--------|--------|--------|--------|
| 5 | 2 | + -> 0 | 0 | 0 |
| 2 | 0 -> + | 3 | + -> 1 | 1 |
| 0 | 1 | 1 -> + | 2 | 2 |
| 1 | 3 | 2 | 3 | 3 |
| 4 -> + -> 4 | 4 | 4 | 4 | 4 |
| 7 | 9 | 9 | 9 | 5 |
| 6 | 8 | 8 | 8 | + -> 6 |
| 3 | 6 | 6 | 6 -> + | 7 |
| 8 | 7 | 7 | 7 | 8 |
| 9 | 5 | 5 | 5 | 9 |

During pass 1, the program chooses an entry at the middle of the list (4) as the *pivot*. The loop on lines 92 through 100 goes down the list and place entries less than 4 starting at the top of the list and downward, and entries greater than 4 starting at the bottom of the list and upward. The 4 is then placed at the remaining spot (between the lesser and greater entries), which spot is the final spot for that pivot.

Pass 2 looks at the partition running from `lo` index 0 to `hi` index 3. The same procedure as described in the previous paragraph is applied to this partition, with 0 being the pivot.

Pass 3 looks at the partition running from `lo` index 1 to `hi` index 3. The same procedure is applied to this partition, with 1 being the pivot.

The final pass, *i.e.*, pass 4, looks at the partition running from `lo` index 5 to `hi` index 9. The same procedure is applied to this partition, with 6 being the pivot. The list is now sorted.