Algorithm for the sort

Member function: partition, parameter: int first, int last

Swap the first and the middle element

Pivot🡨sortingArray[first]

smallIndex🡨first

for index🡨first+1 to last increments by 1

if list[index]<pivot

smallIndex🡨 smallIndex+1

swap smallIndex and index

end if

end for

swap first and smallIndex

return smallIndex

Member function: recQuickSort, parameter: int first, int last

If first<last

pivotLocation 🡨 partition(first, last)

call recQuickSort (first, pivotLocation - 1)

call recQuickSort (pivotLocation + 1, last)

endif

Member function: quicksort

call recQuickSort(0, size of array - 1)