SELECTION SORT:

DECLARE minimum variable

DECLARE position variable

FOR position = 0 AS LONG AS positon is less than vector size LOOP and then INCREMENT position variable

minimum = position

DECLARE j variable

FOR j = position + 1 AS LONG AS j is less than vector size LOOP and then INCREMENT j variable

IF item at index j of vector’s title is less than item at index minimum of vector’s title

minimum = j

ENDIF

ENDFOR

IF minimum is not position

SWAP item at index position of vector and item at index minimum of vector

ENDIF

ENDFOR

PARTITION:

DECLARE low variable and ASSIGN begin argument

DECLARE high variable and ASSIGN end argument

DECLARE pivot variable and ASSIGN middle index between begin and end

DECLARE done and ASSIGN false

WHILE done is false

WHILE item at low index of vector is less than item at pivot index of vector

INCREMENT low

ENDWHILE

WHILE item at pivot index of vector is less than item at high index of vector

DECREMENT high

ENDWHILE

IF low is less than or equal to high

done = true

ELSE

swap item at low index of vector with item at high index of vector

INCREMENT low

DECREMENT high

ENDIF

ENDWHILE

RETURN high

QUICK SORT:

DECLARE mid variable and assign 0

IF begin argument is greater than or equal to end argument

RETURN

ENDIF

ASSIGN mid with return value of PARTITION with arguments of bids vector, begin argument and end argument

CALL quicksort (recursively) with arguments of bids vector, begin argument and mid argument

CALL quicksort (recursively) with arguments of bids vector, mid argument + 1 and end argument