Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# Best: O(n) time | O(1) space
# Average: O(n) time | O(1) space
# Worst: O(n^2) time | O(1) space
def quickselect(array, k):
    position = k - 1
    return quickselectHelper(array, 0, len(array) - 1, position)


def quickselectHelper(array, startIdx, endIdx, position):
    while True:
        if startIdx > endIdx:
            raise Exception("Your algorithm should never arrive here!")
        pivotIdx = startIdx
        leftIdx = startIdx + 1
        rightIdx = endIdx
        while leftIdx <= rightIdx:
            if array[leftIdx] > array[pivotIdx] and array[rightIdx] < array[pivotIdx]:
                swap(leftIdx, rightIdx, array)
            if array[leftIdx] <= array[pivotIdx]:
                leftIdx += 1
            if array[rightIdx] >= array[pivotIdx]:
                rightIdx -= 1
        swap(pivotIdx, rightIdx, array)
        if rightIdx == position:
            return array[rightIdx]
        elif rightIdx < position:
            startIdx = rightIdx + 1
        else:
            endIdx = rightIdx - 1


def swap(one, two, array):
    array[one], array[two] = array[two], array[one]

```