Solution 1

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

// Best: O(n) time | O(1) space
// Average: O(n) time | O(1) space
// Worst: O(n^2) time | O(1) space
func Quickselect(array []int, k int) int {
  return helper(array, 0, len(array)-1, k-1)
}

func helper(array []int, start, end int, position int) int {
  for {
    if start > end {
      panic("This should never happen!")
    }
    pivot, left, right := start, start+1, end
    for left <= right {
      if array[left] > array[right] && array[right] < array[pivot] {
        swap(left, right, array)
      }
      if array[left] <= array[pivot] {
        left += 1
      }
      if array[right] >= array[pivot] {
        right -= 1
      }
    }
    swap(pivot, right, array)
    if right == position {
      return array[right]
    } else if right < position {
      start = right + 1
    } else {
      end = right - 1
    }
  }
}

func swap(one, two int, array []int) {
  array[one], array[two] = array[two], array[one]
}
```