```
def partition (A, begin, end):
    # Partition the range of A that starts at position 'begin'
    # (included) and ends right before position 'end' (excluded).

    # 1. choose a pivot value 'v' at random within A and we switch
    #    it to the last position in A.
    v_pos = random.randint(begin, end-1)
    A[v_pos], A[end-1] = A[end-1], A[v_pos]
    v = A[end-1]

    # 2. loop over the 'begin'--'end' range, with 'q' indicating the
    #    position of the pivot, and therefore the boundary between, on
    #    the left, values that are less than or equal to the pivot,
    #    and on the right, values that are greater than the pivot
    q = begin
    for i in range(begin,end):
        if A[i] <= v:
            A[i], A[q] = A[q], A[i]
            q = q + 1
    # return the final position of the pivot
    return q - 1

def quicksort (A, begin, end):
    # Visualization code: show the focus range between 'begin' and
    # 'end', and if that range needs to be partitioned also shows the
    # pivot value chosen by the partition algorithm.
    for _ in range (0,begin):
        print ('   ',end='')
    for i in range (begin,end):
        print ('%3d' % A[i],end='')
    for _ in range (end,len(A)):
        print ('   ',end='')
    print ('  [', begin, ',', end, ')', sep='', end='')

    # Quick Sort: if the focus range, between 'begin' and 'end'
    # contains two or more elements, then partition that range, and
    # then proceed recursively on the left and right sub-ranges
    # resulting from the partition.
    if end - begin > 1:
        q = partition(A, begin, end)
        print (' q=', q, ' v=', A[q], sep='')
        quicksort (A, begin, q)
        quicksort (A, q + 1, end)
    else:
        print ()
```