Workshop11

Name:________________________________

7.1 Description of quicksort

Listing 1: quicksort

```
QUICKSORT(A,p,r)
  if p < r
       q = PARTITION(A,p,r)
       QUICKSORT(A,p,q-1)
       QUICKSORT(A,q+1,r)
```

Initial call is QUICKSORT($A, 1, n$).

Partitioning

Partition subarray $A[p..r]$ by the following procedure:

Listing 2: partitioning

```
PARTITION(A,p,r)
  x = A[r]
  i = p - 1
  for j = p to r - 1
       if A[j] <= x
            i = i + 1
            exchange A[i] with A[j]
  exchange A[i + 1] with A[r]
  return i + 1
```

- PARTITION always selects the last element $A[r]$ in the subarray $A[p..r]$ as the *pivot*—the element around which to partition.
- As the procedure executes, the array is partitioned into four regions, some of which may be empty: **Loop invariant:**
  1. All entries in $A[p..i]$ are $\leq$ pivot.
  2. All entries in $A[i+1..j-1]$ are $>$ pivot.
  3. $A[r] =$ pivot.

It's not needed as part of the loop invariant, but the fourth region is$A[j..r-1]$, whose entries have not yet been examined, and so we don't know how they compare to the pivot.

1. Using Figure 7.1 pg. 172 as a model, illustrate the operation of PARTITION on the array $A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11 \rangle$.
2. Give a brief explanation of why the running time of PARTITION on a subarray of size n is $\Theta(n)$.
3. Use the loop invariant to prove correctness of PARTITION. Hint: The answer is in the book! Try to do each step withot the book first, then check and correct if necessary.