Huize Shi

A92122910

Feb. 7, 2016

CSE12 HW5

Problem1:

Partition({1, 3, 5, 7, 9, 2, 4, 8}, 0, 7):

A = {1, 3, 5, 7, 9, 2, 4, 8}, x = 8, i = -1

For loop: j = 0; j < 7; j++

Iteration 0:

$A[0] \leq 8$

i = -1 + 1

no swapping

Iteration 1:

$A[1] \leq 8$

i = 0 + 1

no swapping

Iteration 2:

$A[2] \leq 8$

i = 1 + 1

no swapping

Iteration 3:

$A[3] \leq 8$

i = 2 + 1

no swapping

Iteration 4:

$\neg A[4] \leq 8$

Iteration 5:

$A[5] \leq 8$

i = 3 + 1

swap A[4] with A[5]

A = {1, 3, 5, 7, 2, 9, 4, 8}

Iteration 6:

$A[6] \leq 8$

i = 4 + 1

swap A[5] with A[6]

A = {1, 3, 5, 7, 2, 4, 9, 8}

end of loop

exchange A[5+1] with A[7]

A = {1, 3, 5, 7, 2, 4, 8, 9}

return (5 + 1)

Resulting Array:

A = {1, 3, 5, 7, 2, 4, 8, 9}

Problem 2:

a. Finding the minimum value does not require a sorted collection.

Algorithm:

1. Record the first element as min
2. Loop through the array and assign any value that is less than min to min.
3. The resulting min is the minimum value of the array.

Asymptotic Complexity: Θ(n)

b. Finding the maximum value does not require a sorted collection.

Algorithm:

1. Record the first element as max
2. Loop through the array and assign any value that is greater than max to max.
3. The resulting max is the minimum value of the array.

Asymptotic Complexity: Θ(n)

c. Adding each element of the array does not benefit from sorting.

Algorithm:

1. Initialize a running sum to 0
2. Loop through the array and sum += each element.
3. The resulting sum divided by the size of the array is the arithmetic mean.

Asymptotic Complexity: Θ(n)

d. Sorting would be the way to find median since median is defined as the center of a sorted array of integers.

Algorithm (Insertion Sort):

1. Create a nested loop with the outer loop sorting through the array. The inner loop sort from the current value of the outer loop index to 0.
    a. For each element of the outer loop, find the appropriate place for it in the sorted portion using the inner loop.
2. If the array has an odd size, the median is the $\frac{size}{2}$ element of the array.
3. If the array has even size, the median is the average between $\frac{size}{2}$ element and the $\frac{size}{2} - 1$ element.

Asymptotic Complexity: $\Theta(n^2)$

e. Sorting does not benefit finding the mode of a integer array because the sorting algorithm plus the searching algorithm is more complex than just a less efficient searching algorithm.

Algorithm:

1. Initialize 4 variables: num[], freq, tempNum[] and tempFreq.
2. Create a nested for loop each start at the beginning and run to the end
3. Loop through the outer array, record the element as tempNum for each iteration.
    a. For the inter loop, add 1 to tempFreq for every occurrence of tempNum
    b. If tempFrq > freq, then add tempFreq to freq and num to tempNum.
4. Loop through freq to find the top repeated

frequencies then record the index.

5. Num[index] (could be one or more) will be the mode(s).

Asymptotic Complexity: $\Theta(n^2)$

Problem 3:

printInt(5):

5

4

3

2

1

printInt2(5):

1

2

3

4

5