Name: Longchang Cui

ID: 16242D

Course: EN.605.621.81.SP21 Foundations of Algorithms

Programming Assignment #3

a)

We need to create a *MEDIAN-OF-THREE* function to calculate the median value, then return the value and the index. Next, we will use the *MEDIAN-OF-THREE* function to implement the *PARTITION* function.

Pseudo Code

MEDIAN-OF-THREE(A, l, r)   // Calculate the median of three pivots, and return the element and index

```
mid = (l + r - 1) / 2
a = A[l]
b = A[mid]
c = A[r-1]
if a <= b <= c
        return b, mid
if c <= b <= a
        return b, mid
if a <= c <= b
        return c, r – 1
if b <= c <= a
        return c, r – 1
return a, l
```

```
PARTITION_MEDIAN(A, l, r)        // Partition by median
pivot, pivot_index = MEDIAN-OF-THREE(A, l, r)          // Get the median pivot element and index
A[l], A[pivot_index] = A[pivot_index], A[l]            // Swap the first element and pivot element
i = l + 1
for j = l + 1 to r
    if A[j] < pivot:                                   // If the current element is less than pivot, then swap
        A[i], A[j] = A[j], A[i]
        i += 1
A[l], A[i-1] = A[i-1], A[l]
return i - 1
```

```
QUIKSORT_MEDIAN(A, l, r)        // Recursively call the method to sort the array
if l < r
    p = PARTITION_MEDIAN(A, l, r)
    QUICKSORT_MEDIAN(A, l, p)
    QUICKSORT_MEDIAN(A, p + 1, r)
```

b)

The 2-way partition quick sort in the worst case will run $O(n^2)$. But when we execute the median-of-three partition, we guarantee that the selected partition is always the median and the partitions will have the same size. Thus, the average-case and the worst-case will have the same running time.

The *QUIKSORT_MEDIAN* recursive call will be executed $T\left(\frac{n}{2}\right)$ times regardless of average or worst case.

The partitioning method will run $\theta(n)$ time regardless of average or worst case.

Thus,

$$T(n) = 2 * T\left(\frac{n}{2}\right) + \theta(n) = O(nlgn)$$

The average case and the worst case of median-of-three partition quick sort are both $O(nlgn)$.

c)

The *MEDIAN-OF-THREE* will always return the median pivot and the partitions will have the same size even if the input set is already sorted. Thus, we can use the same approach that we used in problem **b**.

The *QUIKSORT_MEDIAN* recursive call will be executed $T\left(\frac{n}{2}\right)$ times in sorted array.

The partitioning method will run $\theta(n)$ time in sorted array.

Thus,

$$T(n) = 2 * T\left(\frac{n}{2}\right) + \theta(n) = O(nlgn)$$

The worst-case of median-of-three quick sort running time in sorted array is $O(nlgn)$.

d)

The implementation of 2-way partition quick sort and the median-of-three quick sort is in the file **Cui_PA3.py**.

e)

i, Testing code is in the file **Cui_PA3.py** from line 44 to 85. The output is in the file **Cui_PA3_Output.txt**

Average-Case

n	10	100	1000	10000
Partitioning execution time	23	734	11843	163861
Sorting execution time	13	144	1467	14810

Worst-Case

n	10	100	1000
Partitioning execution time	45	4260	420195
Sorting execution time	19	198	2017

As we can observe from the above table, the average-case running time of 2-way partition quick sort is $O(nlgn)$ and the worst-case running time is $O(n^2)$

ii, Testing code is in the file **Cui_PA3.py** from line 139 to 184. The output is in the file **Cui_PA3_Output.txt**

Average-Case

n	10	100	1000	10000
Partitioning execution time	10	110	1110	11110
Sorting execution time	21	222	2223	22224

Worst-Case

n	10	100	1000	10000
Partitioning execution time	10	120	1230	12340
Sorting execution time	21	242	2463	24684

Just like we expected on problem **b**, by observing the table above, we can conclude that the average-case running time and the worst-case running time are indeed $O(nlgn)$.

References:

Can anyone give an example for worst case of quick sort if we employ median of three pivot selection? (2016, April 23). Computer Science Stack Exchange. https://cs.stackexchange.com/questions/56377/can-anyone-give-an-example-for-worst-case-of-quick-sort-if-we-employ-median-of-t

P. Kirschenhofer, P. K., H. Prodinger, H. P., & C. Martınez, C. M. (1996). *Analysis of Hoare's FIND Algorithm with Median-of-Three Partition*. Https://Www.Cs.Upc.Edu/. https://www.cs.upc.edu/~conrado/research/papers/rsa-kpm97.pdf

Python: Quicksort with median of three. (2018, June 18). Stack Overflow. https://stackoverflow.com/questions/50912873/python-quicksort-with-median-of-three

Nilsson, S. (n.d.). *Quicksort optimizations explained [complete code]*. Yourbasic.Org. Retrieved April 11, 2021, from https://yourbasic.org/golang/quicksort-optimizations/#3-way-partition

GeeksforGeeks. (2016, December 20). *When does the worst case of Quicksort occur?* https://www.geeksforgeeks.org/when-does-the-worst-case-of-quicksort-occur/