605.621.81.SP19 Foundations of Algorithms

Markham Shofner – Programming Assignment #2

1.

a. *Pseudocode for median of 3 partition*

```
// build the array of three values [a[i], a[j], a[k]], or just pass them in individually
// sort the three values
// chose the middle value of the sorted values
// return the middle value
```

b. *Running time*

The running time of median of three partitioning is liner time since we have no need to nest any loops. All we are doing is taking 3 input values and finding and outputting the middlest value.

c. *Running time on a sorted input set*

The running time on a sorted input is VASTLY superior for median of 3 than the approach of picking either the first of last element in the set as the pivot. This is because those would reduce the problem space at a much slower pace than the median of 3 which would perfectly pick the partition each time [right in the middle], so we are running an optimal quicksort which means: Log(n)

d. *Implementation*

Input Type	**First partition move count**	**Median of 3 move count**
Ascending500	3,864	892
Reverse500	3,864	892
Random500	3,624	1,060
Ascending10000	79,864	14,332
Reverse10000	79,864	14,332
Random10000	78,904	20,408

- I expected the Median of 3 to perform better than the First Partition, especially on the ascending/descending sets, but not by quite this large of an amount. Pretty eye opening difference in efficiency and cost between the two.