

Runtime of Lab 4

1) Input, create int n. $\Theta(1)$

2) Generate n numbers and store in array a $\Theta(n)$

3) Print array.

4) Create int k $\Theta(1)$

5) Find median of array

I used quickSelect to find the $(n/2)$ least number = median

→ average case $\Theta(n)$

6) $(a[i] - \text{median})$ values in array diff.

This has runtime $\Theta(n)$.

Have to check each index in n to fill diff.

7) medianNeighbors { modified quickSelect

 pivotIndex = partitionAbs(a, low/high) + 1;

 if ($k < \text{pivotIndex}$)

 a = check subarray left of pivot

 else if ($k > \text{pivotIndex}$)

 a = check subarray right of pivot

 else // answer

 a = subarray(0, k)

 return a;

}

$n, n/2, n/4, n/8$

$n(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \dots)$

$1 + \boxed{\quad} < 2$

$2n = n$

Runtime: $n + 2n$
partition subarray

$\Theta(n)$

8) Shift k values back to original value

At worst $n-1$ values = $\Theta(n)$

9) Print answer

Runtime: $n + n + n + n + n + n + n$

lots of steps with n runtime

$\Theta(n)$

partitionAbs { Iterates $(n-1)$
runtime $\Theta(n)$

 Set pivot to last element

 left = 0;

 right = pivot - 1;

 if (length == 1) {

 return pivot;

 }

 while (left < right) {

 while ($\text{abs}(a[\text{left}]) < \text{abs}(a[\text{pivot}]) \ \&\& \ \text{left} < \text{right}$) {

 left++;

 }

 right while loop

 - same conditions but

 - right--;

 swap left and right if frozen

 }

 final pivot swap, return pivotIndex;