Jacob Hansen
COSC 3020
Lab 3
9/25/2019

Looking over the code in Lab3.js, this is the time complexity of each step.

Line(s)	Description	Time Complexity
10	Set variable	1
11	If array is empty, stop	1
15	If array has 1 entry, return value of that entry	1
20-22	Calculate start/endpoints of smaller arrays to be used in next step	1
25-27	Add to current sum the sum of 1/3 of the array, recursively	1 + 3T(n/3)
29	No more steps, return value of sum	1

We ignore all constant times, leaving only the recursive part.

$$1 + 3T\left(\frac{n}{3}\right)$$

We remove the constant from this as well.

$$3T\left(\frac{n}{3}\right)$$

Considering the three cases, n = 0 and n = 1 are constant time with n > 1 being not constant.

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ 3T\left(\frac{n}{3}\right) & \text{if } n > 1 \end{cases}$$

We now have the recurrence relation. Now we use substitution to find Θ.

$$T(n) = 3T\left(\frac{n}{3}\right)$$

$$T(n) = 3\left(3T\left(\frac{n/3}{3}\right)\right) = 9T\left(\frac{n}{9}\right)$$

$$T(n) = 27T\left(\frac{n}{27}\right)$$

$$T(n) = 3^iT\left(\frac{n}{3^i}\right)$$

For i = lg n

$$nT\left(\frac{n}{n}\right) = nT(1) = n$$

The result is a time complexity Θ = n, or linear.