Hands: On 6

3. The average runtime complexity of non-random pivot version of quicksort can be mathematically derived as follows:

→ In quicksort, we always choose the middle element of the array as pivot.

→ We go through the array once to partition it around the pivot we selected then it takes $O(n)$ time. After this, we sort left and right subarrays recursively.

Lets assume, pivot divides array into two parts equally (mostly), each recursive call operates on a subarray of size (approx) $= n/2$

⇒ This leads to a recurrence relation of the form:

$$T(n) = O(n) + 2T(n/2)$$

(∵ $2 = 2^1$, $a = b^d$)

using master theorem, $a = 2$, $b = 2$, $d = 1$

$$\therefore T(n) = \Theta(n \lg n)$$

Hence, the average runtime complexity of non-random pivot version of quicksort is $O(n \lg n)$