Question 7

(a)

Let $T(n)$ be the cost of the algorithm. Then, since we recursively operate on the left and right halves of the input array, and we iterate through all n elements per recursive call, we derive the following recurrence relation:

$$T(n) = \begin{cases} 1, & n = 1 \\ T(n/2) + n, & n > 1 \end{cases}$$

The cost of the base case is 1, since we simply return the single element.

(b)

Let $a = 2, b = 2, c = 1, d = 1$, and $f(n) = n$. Then, $a, c \geq 1, b > 1$, and $d \geq 0$. So we may proceed to use the Master Theorem to derive the cost of the algorithm.

We see that $\log_b(a) = \log_2(2) = 1$, so $n^{\log_2(2)} = n^1$. Then $f(n) \in \Theta(n)$, so case 2 gives us:

$$T(n) \in \Theta(n \log(n))$$