

Problem 1: -

Method 1

$$T(n) = T(n - 1) + c \quad \text{if } n > 1$$

$$T(n) = T(n - 1) + c$$

$$T(n) = T(n - 2) + c + c$$

$$T(n) = T(n - 3) + c + c + c$$

Inputting value 1 to value k

$$T(n) = T(n - k) + k * c$$

$$n - k = 1 \Rightarrow k = n - 1$$

$$T(n) = T(1) + (n - 1) * c$$

$$T(n) = O(1) + n * c - c$$

$$T(n) = O(n)$$

Therefore, the time complexity is $O(n)$

Method 2

$$T(n) = T(n/2) + c \quad \text{if } n > 1$$

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

$$T(n) = T(n/2^2) + c + c$$

$$T(n) = T(n/2^3) + c + c + c$$

$$T(n) = T(n/2^k) + k * c$$

$$n/2^k = 1 \Rightarrow 2^k = n \Rightarrow k = \log^n$$

Inputting value 1 to value k

$$T(n) = T(1) + c * \log^n$$

$$T(n) = O(1) + c * \log^n$$

$$T(n) = O(\log^n)$$

Therefore, the time complexity is $O(\log^n)$

Method 3

$$T(n) = T(n/2) + T(n/2) + c = 2 * T(n/2) + c \quad \text{if } n > 1$$

$$T(n) = 2 * T(n/2) + c$$

$$T(n) = 2^2 * T(n/2^2) + c + c$$

$$T(n) = 2^3 * T(n/2^3) + c + c + c$$

$$T(n) = 2^k * T(n/2^k) + k * c$$

$$n/2^k = 1 \Rightarrow 2^k = n \Rightarrow k = \log n$$

Inputting value 1 to value k

$$T(n) = n * T(1) + c * \log n$$

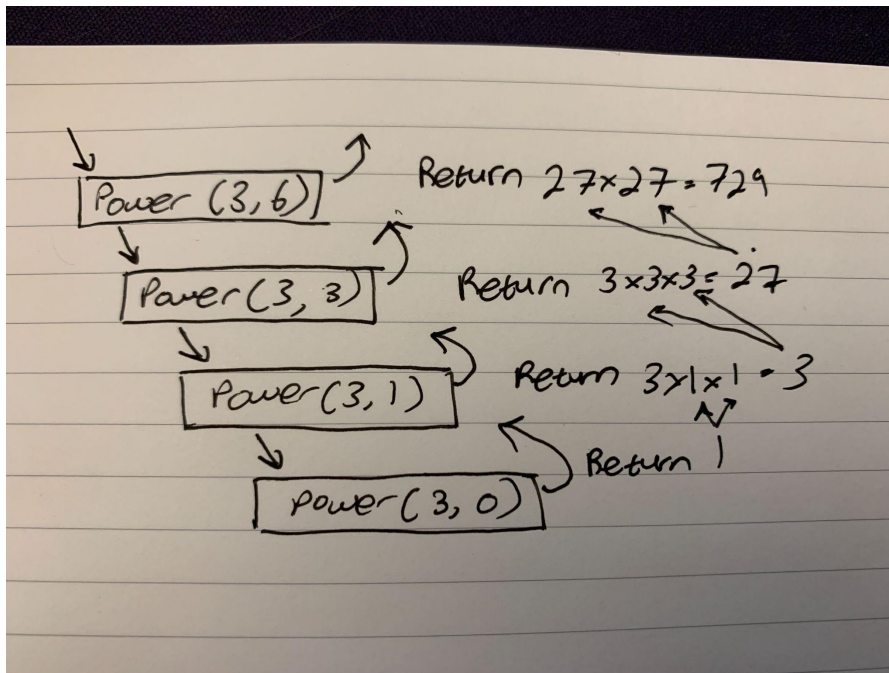
$$T(n) = n + c * \log n$$

$$T(n) = O(n)$$

Therefore, the time complexity is $O(n)$

Problem 2: -

Question 1



Question 2

