

Q2.

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 3T(n/3) + n - 1 & \end{cases}$$

$$\begin{aligned} T(n/3) &= n-1 + 3 \left(\frac{n}{3} - 1 + 3T(n/9) \right) \\ &= n-1 + n-3 + 9T(n/9) \end{aligned}$$

$$\begin{aligned} T(n/9) &= n-1 + n-3 + 9 \left(\frac{n}{9} - 1 + 3T(n/27) \right) \\ &= n-1 + n-3 + n-9 + 27T(n/27) \end{aligned}$$

$$= n-3^0 + n-3^1 + n-3^2 + 3^3 T(n/3^3)$$

$$= n-3^0 + n-3^1 + n-3^2 + \dots + 3^k T(n/3^k)$$

$$= k \cdot n - (3^0 + 3^1 + 3^2 + \dots + 3^{k-1}) + 3^k T(n/3^k)$$

$$\frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow k = \log_3 n$$

$$n \log_3 n - \left(\sum_{i=0}^{\log_3 n - 1} 3^i \right) + 3^{\log_3 n} T(1)$$

$$\approx n \log_3 n - n \cancel{1} \cancel{1}$$

$$\approx n \log_3 n$$

$$\rightarrow \text{time complexity} : O(n \log_3 n)$$

$$\begin{aligned} \sum_{i=0}^{\log_3 n - 1} 3^i &= 3^{\log_3 n} - 1 \\ &= 3^{\log_3 n} - 1 \\ &= n - 1 \end{aligned}$$

Based on the calculation that is done in the previous part, the time complexity of 3-way merge sort is $O(n \log_3 n)$ and time complexity in 2-way merge sort is $O(n \log_2 n)$. by comparing the bases in logs we can see that 2-way has smaller base in log than 3-way and generally, smaller bases in log functions show better time complexity, because in fact these bases show the number of time that we are recursively calling function. In other words. Smaller base in log function, shows a smaller number of recursive steps (calls). So, in larger amount of data, 3-way merge sort creates 3 recursive calls each time, while 2-way merge sort makes 2 recursive calls at each time. Thus, most of the time 2-way merge sort is more efficient.

In-place merge sort algorithm:

Based on what I understand its not possible to have a $O(1)$ space complexity if we use array in the merge sort. It will use some extra space to store the subarrays, so the space complexity might be less with using other algorithms, but it can not be $O(1)$. But if we use a different data structure like a linked list it is possible to have a $O(1)$ space complexity for in-place merge sort.

The algorithm with using linked list:

1. Split the linked list in two halves.
2. Sorting each half of the linked list by recursively calling merge sort on each half of the linked list
3. Apply the merge function to merge the two sorted linked lists