1) In this question basicly we can think as each time the maxlength dividing to so $\boxed{\log_2 \text{length}}$ give us the result

but we need to convert it decrease and conquer algorithm we can subtruct the $2^1, 2^2, 2^3, \ldots, 2^n$ one by one from the lenghth of wire the n gives us the result

$$T(n) = T(n - 2^i) + 1 \quad , \quad n<0 \quad 0$$

$n = 7$

$$\left.\begin{aligned} T(7) &= T(5) + 1 \\ T(5) &= T(1) + 1 \\ T(1) &= T(<0) + 1 \end{aligned}\right\} 3$$

$$T(n) \begin{cases} 0 & ; n<0 \\ T(n-2^i)+1 & ; n>0 \end{cases}$$

so $\boxed{T(n) \in \Theta(\log n)}$

---

2)

$$T(n) = 2T\left(\frac{n}{2}\right) + 1$$

$a = 2$
$b = 2$
$d = 0$

$a \, ? \, b^d$
$2 \, ? \, 2^0$
$2 > 1$

$$T(n) \in \Theta(n^{\log_b a})$$

$$\boxed{\in \Theta(n)}$$

- It is dividing 2 each recursive call until length of array > 1 and when it back track it compare the values and finding max and min values.

---

3) We are doing quick select algorithm for finding nth smallest and it is decrease and conquer using lomuto partition

$$T(n) = T\left(\frac{n}{2}\right) + n$$

$a = 1$
$b = 2$
$d = 1$

$a \quad b^d$
$1 < 2^1$

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

4) We are doing a merge sort and when we merge the two list increasing the revers pair counter so the time analysis.

$$T(n) = \underbrace{2T(n/2)}_{\text{dividing array}} + n$$

concanate the dividing array and increase the counter

$a=2$ $b=2$ $d=1$ $2 = 2^1$

$$\boxed{T(n) = (n \log n)}$$

---

5) a) Brute force :

Basic multiply a to a n times.

$$\boxed{T(n) = \Theta(n)}$$

Divide & Conq:

$$T(n) = T(n/2) + 1$$

$a=1$
$b=2$
$d=0$

$1 = 2^0$

$$T(n) = \Theta(n^0 \log n)$$

$$\boxed{= \Theta(\log n)}$$

Muhammed Bedir ULUCAY
1801042697