Divide and Conquer

What is the solution to the recurrence $T(n) = T(n/2) + \Theta(n)$? (Let $T(1) = b$, where b is a constant)

O(n)

Given the following recurrence relation:

$T(1) = a,$

$T(n) = qT(n/2) + bn, \ q>2$

a. T(n) is $O(qn)$
b. T(n) is $O(qn^2)$
c. T(n) is $O(q\log_2(n))$
d. T(n) is $O(qn\log_2(n))$
e. T(n) is $O(n\log(q))$
f. T(n) is $O(q^{\log(n)})$
g. T(n) is $O(n^{\log(q)})$
h. None of above

In the recurrence tree for Merge-Sort algorithm, the work done at any level 'k' of the tree is equal to half the amount of work at level k-1.

Do you agree with the above statement? **YES / NO** _____No______ If Yes, **prove** it. If No, then explain why. Same work i.e. O(n) at each level.

Solve the following recurrence relation to determine T(n).

$T(1) = b_1$
$F(1) = b_2$
$F(n) = F(n-1) + c$
$T(n) = 3T(\frac{n}{2}) + F(n)$, where b_1, b_2 and c are constants.

Solve F(n) first. $F(n) = F(n-1) + c$

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

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

$\vdots$

$F(n) = F(n-K) + Kc$ Let $K = n-1$

$F(n) = F(1) + (n-1)c = b_2 + (n-1)c$

Since $F(n)$ is $O(n)$, we can simplify it as $F(n) = c_1 n$, where c_1 i[s] a constant

$$T(n) = 3T\left(\frac{n}{2}\right) + c_1 n$$

$$T(n) = 3\left(3T\left(\frac{n}{4}\right) + c_1\frac{n}{2}\right) + c_1 n$$

$$T(n) = 3^2\, T\left(\frac{n}{4}\right) + 3c_1\frac{n}{2} + c_1 n$$

$$T(n) = 3^2\left(3T\left(\frac{n}{8}\right) + c_1\frac{n}{4}\right) + 3c_1\frac{n}{2} + c_1 n$$

$$T(n) = 3^3\, T\left(\frac{n}{2^3}\right) + 3^2 c_1\frac{n}{2^2} + 3c_1\frac{n}{2} + c_1 n$$

$\vdots$

$$T(n) = 3^K\, T\left(\frac{n}{2^K}\right) + \left(\left(\frac{3}{2}\right)^{K-1} + \left(\frac{3}{2}\right)^{K-2} + \dots + \frac{3}{2} + 1\right)c_1 n$$

$$T(n) = 3^{\log_2(n)}\, T(1) + c_1 n\left(\frac{\left(\frac{3}{2}\right)^K - 1}{\frac{3}{2} - 1}\right) \qquad \frac{n}{2^K} = 1 \quad K = \log n$$

$$T(n) = 3^{\log_2(n)} \cdot b_1 + 2c_1 n\left(\left(\frac{3}{2}\right)^{\log_2 n} - 1\right)$$

$$= 3^{\log_2(n)}\, b_1 + (2c_1)(n)\left(\frac{3^{\log_2(n)}}{n} - 1\right)$$

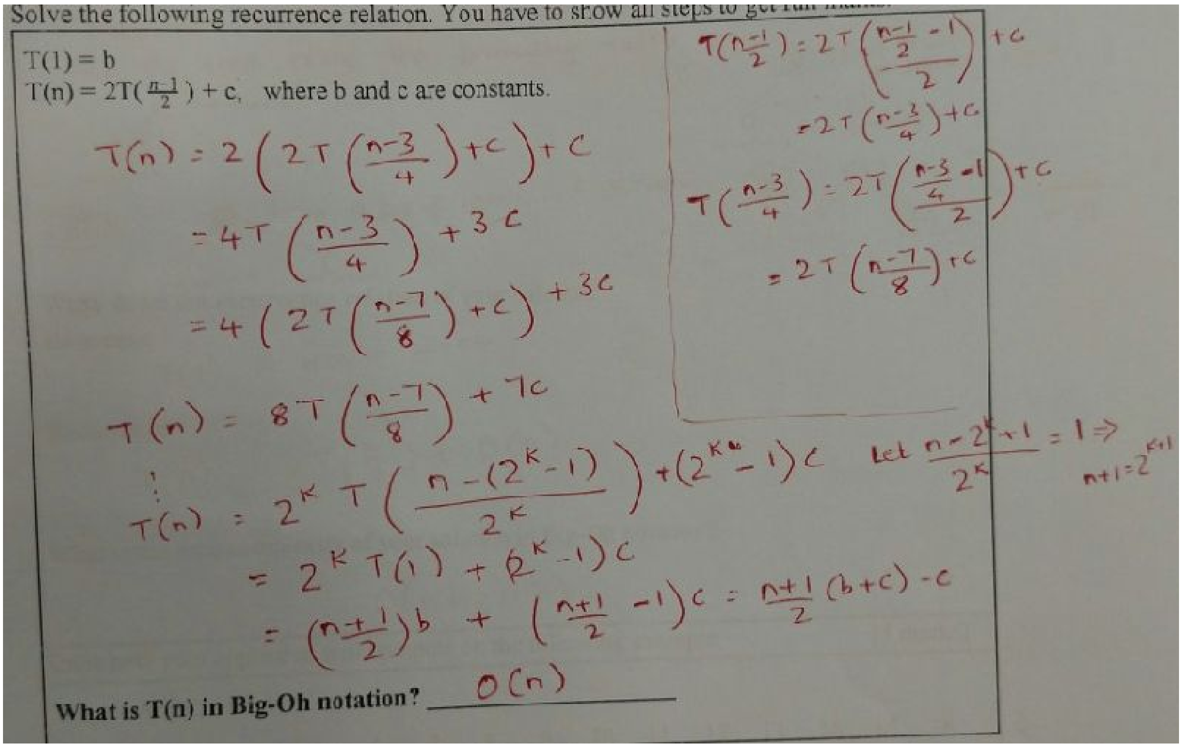
$$= 3^{\log_2(n)}(b_1 + 2c_1) - 2c_1 n$$

$3^{\log_2(n)} = n^{\log_2(3)}$ (Formula was given on the last page)

$T(n)$ is $O\left(n^{\log_2(3)}\right)$

Solve the following recurrence relation. You have to show all steps to get full marks.

T(1) = b
T(n) = 2T($\frac{n-1}{2}$) + c,   where b and c are constants.



Solve the following recurrence relation. You have to show all steps to get full marks.

T(1) = b
T(n) = 2T($\frac{n-1}{2}$) + c,   where b and c are constants.

$$T\left(\frac{n-1}{2}\right) = 2T\left(\frac{\frac{n-1}{2}-1}{2}\right) + c$$
$$= 2T\left(\frac{n-3}{4}\right) + c$$
$$T\left(\frac{n-3}{4}\right) = 2T\left(\frac{\frac{n-3}{4}-1}{2}\right) + c$$
$$= 2T\left(\frac{n-7}{8}\right) + c$$

$$T(n) = 2\left(2T\left(\frac{n-3}{4}\right) + c\right) + c$$
$$= 4T\left(\frac{n-3}{4}\right) + 3c$$
$$= 4\left(2T\left(\frac{n-7}{8}\right) + c\right) + 3c$$
$$T(n) = 8T\left(\frac{n-7}{8}\right) + 7c$$
$$\vdots$$
$$T(n) = 2^k T\left(\frac{n-(2^k-1)}{2^k}\right) + (2^k-1)c \qquad \text{let } \frac{n-2^k+1}{2^k} = 1 \Rightarrow n+1 = 2^k$$
$$= 2^k T(1) + (2^k-1)c$$
$$= \left(\frac{n+1}{2}\right)b + \left(\frac{n+1}{2} - 1\right)c = \frac{n+1}{2}(b+c) - c$$

What is T(n) in Big-Oh notation? O(n)

What is T(n) in Big-Oh notation? _______O(n)____________