Find recurrence equation and big-oh for the nodes at level n in binary tree

Note - In this problem, n is being used differently from other problems in that it is not the number of nodes in the tree, it is the level (what previously was called k).

Recurrence equation:

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

Consider n>1:

```
T(n) = 1 + T(n-1)          +         T(n-1)
           \_____/                   \_____/
    1 + T(n-2)  +  T(n-2)            1 +   T(n-2)  +  T(n-2)
        \____/     \____/                  \____/     \____/
 1+T(n-3)+T(n-3)  1+T(n-3)+T(n-3)   1+T(n-3)+T(n-3)  1+T(n-3)+T(n-3)
          ...
```

Each time, at each level of replacement, we get a one (really O(1)) appearing a power of 2 times. At the end, T(1) appears 2^(n-1) times.

$$= 1*(2^{0} + 2^{1} + 2^{2} + 2^{3} + ... + 2^{n-2}) + 2^{n-1}$$

$$= \left(\sum_{i=0}^{n-1} 2^{i} \right) = 2^{n} - 1 \qquad \text{So } T(n) = O(2^{n}).$$