Substituting the given relation recursively, we have:

\[ T(n) = \frac{3}{4} T(n/2) + n = \frac{3}{4} \left( \frac{3}{4} T(n/4) + n/2 \right) + n = \frac{3}{4} \left( \frac{3}{4} \left( \frac{3}{4} T(n/8) + n/4 \right) + n/2 \right) + n \]

If we assume it takes \( h \) steps to get to \( T(1) \), we have:

\[ T(n) = \frac{3^h}{4} T(1) + \sum_{i=0}^{h} \left( \frac{3}{4} \right)^i n = 0 + c \cdot n = O(n) \]

2

\[
\begin{array}{c}
T(n) \\
| \\
T(n/2) \\
| \\
\vdots \\
| \\
T(0)
\end{array}
\]

In the above tree, the costs for each node at each level starting from root is \( 2^n, 2^{n/2}, 2^{n/4}, \ldots, 2^2 \). If we add them up we will have \( T(n) = \sum_{i=1}^{\log_2 n} 2^i \). Notice that the first term in this sigma is bigger than all other terms added up. So \( T(n) = O(2^n) \).
In the above tree, the costs for each node at each level starting from root is \(n^2, (n-1)^2, (n-2)^2, \ldots, 1^2\). If we add them up we will have \(T(n) = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = O(n^3)\).

In the above tree, the costs of each node at each level starting from root is \(n/4, n/4^2, n/4^3, \ldots, 1\). Since there are \(4^i\) number of nodes at level \(i\) (considering root as level 0), we can calculate the aggregate cost for all nodes within a level by multiplying the number of nodes to the cost of each node. Thus, starting from root we will have the following costs: \(1 \cdot \frac{n}{4}, \frac{n}{4^2} \cdot 4, \frac{n}{4^3} \cdot 4^2, \ldots, \frac{n}{4} \cdot 1\). If we add them up we will have \(T(n) = (\log_4(n)) \cdot n/4 = \frac{n \log_4(n)}{4} = O(n \log(n))\).

In the above tree, the costs of each node at each level starting from root is \(n+1, n/2 + 1, n/4 + \ldots\).
1, \ldots, 2. Since there are $2^i$ number of nodes at level $i$ (considering root as level 0), we can calculate the aggregate cost for all nodes within a level by multiplying the number of nodes to the cost of each node. Thus, starting from root we will have the following costs: $1 \cdot (n + 1), 2 \cdot \left( \frac{n}{2} + 1 \right), 4 \cdot \left( \frac{n}{4} + 1 \right), \ldots, n \cdot 2$. Let’s add them up to get the whole cost by all nodes together.

$$T(n) = \sum_{i=1}^{\log_2(n)} 2^i \left( \frac{n}{2^i} + 1 \right) = \sum_{i=1}^{\log_2(n)} n + 2^i = n + \sum_{i=1}^{\log_2(n)} 2^i = n \log_2(n) + n - 1 = \mathcal{O}\left(n \log(n)\right).$$