We analyzed the computational complexity of the algorithm BUILD-HEAP(A) as follows:
For a heap of height $h$ that contains $n$ elements, we have $O(n)$ calls to the algorithm HEAPIFY().
HEAPIFY() has computational complexity $O(h) = O(\log_2 n)$.
Therefore, we have $O(n \log_2 n)$ for BUILD-HEAP(A).

A more careful analysis takes into consideration the computational complexity of HEAPIFY() for a node with depth $d$, and the number of nodes at depth $d$.
Write a summation that expresses the computational complexity of BUILD-HEAP(A) in terms of $h$ and $d$.
You do not need to solve the summation.
Note: Such a summation yields a tighter bound of $O(n)$ for the computational complexity of BUILD-HEAP(A).

Answer (summation):

The number of nodes at depth $d$ is $2^d$.
For a node at depth $d$, HEAPIFY() performs $O(h - d)$ operations (this is the distance from the node to a leaf).
Overall, $\sum_{d=0}^{h} 2^d (h - d)$