C-3.7 Hint: You will need to augment $T$, adding a new field to each internal node and ways of maintaining this field during updates.

Solution: For each node of the tree, maintain the size of the corresponding subtree, defined as the number of internal nodes in that subtree. While performing the search operation in both the insertion and deletion, the subtree sizes can be either incremented or decremented. During the rebalancing, care must be taken to update the subtree sizes of the three nodes involved (labeled $a$, $b$, and $c$ by the restructure algorithm).

To calculate the number of nodes in a range $(k_1, k_2)$, search for both $k_1$ and $k_2$, and let $P_1$ and $P_2$ be the associated search paths. Call $v$ the last node common to the two paths. Traverse path $P_1$ from $v$ to $k_1$. For each internal node $w \neq v$ encountered, if the right child of $w$ is in not in $P_1$, add one plus the size of the subtree of the child to the current sum. Similarly, traverse path $P_2$ from $v$ to $k_2$. For each internal node $w \neq v$ encountered, if the left child of $w$ is in not in $P_2$, add one plus the size of the subtree of the left to the current sum. Finally, add one to the current sum (for the key stored at node $v$).

C-3.8 Hint: Consider adding a small subtree of size $n^{1/2}$ to a binary search tree that is otherwise of good height.

C-3.9 Hint: Consider augmenting each key, $k$, by choosing a random number in the range from 1 to $n$ for each inserted pair, and using $k$ together with the associated random number to guide insertions.

C-3.10 Hint: Think first about how you can determine the number of 1’s in any row in $O(\log n)$ time.

Solution: To count the number of 1’s in $A$, we can do a binary search on each row of $A$ to determine the position of the last 1 in that row. Then we can simply sum up these values to obtain the total number of 1’s in $A$. This takes $O(\log n)$ time to find the last 1 in each row. Done for each of the $n$ rows, then this takes $O(n \log n)$ time.

C-3.11 Hint: Treat the cases $i = 1$ and $i = n$ separately.

Solution: For the upper bound,

$$H_n = 1 + \sum_{i=2}^{n} \frac{1}{i} \leq 1 + \int_{x=1}^{n} \frac{dx}{x} = 1 + \ln n.$$ 

For the lower bound,

$$H_n \geq \sum_{i=1}^{n-1} \frac{1}{i} \geq \int_{x=1}^{n} \frac{dx}{x} = \ln n.$$

C-3.12 Hint: Write out the terms of the difference, $H_n - H_{n/2}$. How many terms are there?