Let \( a_1, a_2, \ldots, a_n \) be a sequence of real numbers, for \( n \geq 1 \). A \texttt{SeqSet} is an ADT which stores the sequence and supports the following operations:

- \texttt{PartialSum} \((S, m)\): return \( \sum_{i=1}^{m} a_i \), the partial sum from \( a_1 \) to \( a_m \) (\( 1 \leq m \leq n \)).
- \texttt{Change} \((S, i, y)\): change the value of \( a_i \) to a real number \( y \).

Design a data structure that implements \texttt{SeqSet}, using an \textbf{augmented AVL tree}. The worst-case running time of both \texttt{PartialSum} and \texttt{Change} must be in \( O(\log n) \). Describe your design by answering the following questions.

(a) What is the key of each node in the AVL tree? What other attributes are stored in each node?

(b) Write the pseudo-code of your \texttt{PartialSum} operation, and explain why your code works correctly and why its worst-case running time is \( O(\log n) \). Let \( S.\text{root} \) denote the root node of the AVL tree. \textbf{Hint:} This pseudo-code can be very similar to that of an operation in the textbook, which one is it?

(c) Describe in clear English how your \texttt{Change} operation works, and explain why it runs in \( O(\log n) \) time while maintaining the attributes stored in the nodes of the AVL tree.