Problem 1.

(a) Starting with an empty binary search tree, give a sequence of \( n \) insertion operations that result in a tree of height \( \Omega(n) \).

(b) Starting with an empty binary search tree, give a sequence of \( n \) insertion operations that result in a tree of height \( O(\log n) \).

Problem 2. In Section 12.4 of the book it is shown that the expected height of a randomly built binary search tree on \( n \) distinct keys is \( O(\log n) \). That is, if we start with an empty tree, and we insert \( n \) distinct elements in a random order, then the expected height of the tree is \( O(\log n) \). Here, by “random order” we mean an order chosen uniformly at random from the set of all possible orderings of the \( n \) keys. Also note that since the ordering is random, the height of the resulting tree is a random variable. In other words, the above result states that the expectation of this random variable is \( O(\log n) \).

Use the above fact to construct a randomized algorithm for sorting \( n \) elements, with expected running time \( O(n \cdot \log n) \).

Hint: You don’t need to use the proof of the above statement about the expected height from Section 12.4. It is enough to assume that the statement is true.