Problem 1. In the class, we proved that if \( f(h) \) denotes the smallest number of nodes in a balanced binary tree of height \( h \), it must hold that

\[
f(h) = 1 + f(h - 1) + f(h - 2).
\]

Give a balanced binary tree of height 6 with \( f(6) \) nodes.

Problem 2. Let \( T \) be a binary tree of \( n \) nodes. For each node \( u \) of \( T \), define its count as the number of nodes in its subtree (remember that the subtree includes the node itself). Describe an algorithm to compute the counts of all the nodes in \( T \) (you can assume that each node has reserved a memory cell for you to store the count). Your algorithm must terminate in \( O(n) \) time.

Problem 3. Let \( T \) be a binary search tree (BST) of on a set \( S \) of \( n \) integers. Let \( x \) and \( y \) be two integers in \( S \). Describe an algorithm to find the lowest common ancestor \( A \) of the nodes in \( T \) storing \( x \) and \( y \), respectively. If \( A \) is at level \( \ell \) (recall that the root is at level 0), your algorithm must finish in \( O(1 + \ell) \) time.

Problem 4. Let \( T \) be a binary search tree (BST) of on a set \( S \) of \( n \) integers. Describe an \( O(\log n + k) \)-time algorithm to answer the following query: given an interval \([a, b]\), report all the integers of \( S \) that fall in \([a, b]\). Here, \( k \) is the number of integers reported.

Problem 5. Let \( S \) be a set of \( n \) key-value pairs of the form \((t, v)\). Denote by \( m \) the number of distinct keys in all the pairs of \( S \). Describe a data structure to support the following queries efficiently: given an interval \([a, b]\), report all the pairs \((t, v)\) in \( S \) such that \( t \in [a, b] \). Your structure must use \( O(n) \) space, and answer a query in \( O(\log m + k) \) time, where \( k \) is the number of pairs reported.