1. (25 points) List the following 8 functions according to their growth rate:

\[ n^{1000}, (n \log n)^{999}, n!, n^{\log(n)}, n^{\log(\log(n))}, 3^{2n}, n^n, (\log n)^n \]

2. (25 points) Display the Red-Black trees after inserting each of the following numbers in the given order: 1, 3, 6, 2, 4, 5 (using double circles for black nodes and single circle for red nodes).

3. (25 points) Given a balanced binary search tree \( T \) of \( n \) nodes and a number \( i, 1 \leq i \leq n \), please provide an algorithm (in pseudocode) to return the node in \( T \) which contains the \( i \)th smallest key. If each node contains the size of the subtree rooted by the node, how to use this information in your algorithm? Please provide the complexity of both algorithms.

4. (25 points) Let \( T \) be a rooted binary tree with more than one node. The degree of any node \( x \) in \( T \) is the number of nodes connecting to \( x \) (as its children or parent). A node \( y \) of \( T \) is said to be a core node if there is no path of length two or less from \( y \) to a node \( x \) whose degree is one. Please design an efficient algorithm that identifies all core nodes of \( T \).