Problem 1. Prove that, if a node in a binary-search-tree has two children, its successor has no left child and its predecessor no right-child.

Problem 2. Most graph algorithms that take an adjacency-matrix representation as input require time $\Omega(|V|^2)$, but there are some exceptions. Show how to determine whether a directed graph $G$ contains a universal sink a vertex with in-degree $|V| - 1$ and out-degree 0 in time $O(V)$, given an adjacency matrix for $G$. Pseudocode is required for this problem. Also discuss correctness and analyze the running time.

Problem 3. **Knapsack** is the following problem: Given integers $K$ and $B$, and items \{1, 2, \ldots, n\} each with integer size $s_i$ and integer profit $p_i$, find a subset $D \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in D} s_i \leq B$ and $\sum_{i \in D} p_i \geq K$. As an aside, the common optimization problem is: we attempt to maximize $K$, the total profit, with a bound $B$ on the total size of items one can pick (the set $D$). All the numbers are written in binary. Prove that **Knapsack**, as a language or decision problem, is NP-hard. The reduction must come from a problem proved NP-hard in the textbook.