Problem 1. Consider the problem of determining whether an arbitrary sequence \( \langle x_1, x_2, \ldots, x_n \rangle \) of \( n \) numbers contains repeated occurrences of some number. Show that this can be done in \( O(n \log n) \) time, where \( \log n \) stands for \( \log_2 n \).

Pseudocode is required. Do analyze the running time.

Problem 2. We say that a digraph \( G = (V, E) \) is half-connected iff for all \( u, v \in V \), there exists either a directed path from \( u \) to \( v \) or a directed path from \( v \) to \( u \). Give an \( O(|V| + |E|) \)-time and space algorithm to determine if a given digraph (adjacency lists representation) is half-connected.

Pseudocode is required. Do analyze the running time and do prove that your algorithm is correct.

Problem 3. Suppose we wish not only to increment a binary number, but also to reset it to zero (i.e., make all bits in it 0). Counting the cost to examine or modify a bit as 1, show how to implement a binary number as an array of bits so that any sequence of \( n \) INCREMENT and RESET operations costs \( O(n) \) on an initially zero number. Do analyze the running time.

Hint: Keep a pointer to the high-order 1.