Instructions: There are six questions in this exam, each containing subproblems. Please answer legibly and thoroughly.

1. **Short questions.**
   a. Explain why all the comparison-based sorting algorithms must run in \( \Omega(n \log n) \) time on an input of \( n \) numbers.
   b. Suppose we are given a sequence \( S \) of \( n \) integers in the range \([0, n^2 - 1]\). Describe a simple method for sorting \( S \) in \( O(n) \) time.
   c. Why does question (b) not contradict question (a)?
2. Given a sorted array of distinct integers $A[1\ldots n]$. We are interested in determining if there is an index $i$ in the array so that $A[i] = i$. Of course, one can simply scan the array to solve the problem. We want to do better.

a. Design an $O(\log n)$-time algorithm that solves the problem. (Hint: Keep in mind that the elements of $A[1\ldots n]$ are distinct integers and already sorted.)

b. Why is your algorithm correct, and why is its runtime $O(\log n)$?
3. **The Minimum Coverage Problem** Let $L$ be a set of $n$ line segments on the $x$-axis. The $i$th line segment is $[l_i, r_i]$, where $l_i$ and $r_i$ denote the left and right endpoints of the line segment respectively. For some positive number $M$, we want to cover the segment $[0, M]$ completely using the fewest number of line segments from $M$.

a. **Warm up!** Let $L = \{[-1.5, 1.5], [-4, 2], [2, 3], [3, 5], [1, 4.5]\}$ and $M = 4.2$. What is the fewest number of line segments from $L$ that cover $[0, 4.2]$. What are they?

b. Given a set $L$ and a positive number $M$, solve the Minimum Coverage Problem using a greedy algorithm. In particular, if the line segments in $L$ can cover $[0, M]$, the algorithm should output the fewest number of line segments that can do so; otherwise, it should indicate that the line segments from $L$ cannot do so.

c. Argue why your answer is correct. What is its runtime?

a. Let $G$ be an undirected graph. Describe an algorithm that outputs "yes" if $G$ has a cycle and "no" otherwise. What is its runtime?

b. Let $G$ be a directed graph. Describe an algorithm that outputs "yes" if $G$ has a directed cycle and "no" otherwise. What is its runtime?
5. *True or false.* If true, please provide a short proof. If false, please illustrate with a counterexample.

a. If the edge weights in $G$ are unique, then $G$ has a unique minimum spanning tree.

b. If $G$ has a unique minimum spanning tree, then the edge weights in $G$ are unique.
6. In cases where there are several different shortest paths between two nodes, the most convenient of these paths is often the one with fewest edges. For instance, if nodes represent cities and edge lengths represent costs of flying between cities, there may be many ways to get from city \( s \) to city \( t \) all of which have the same cost. The most appealing of these alternatives is the one which involves the fewest stopovers. Accordingly, for a specific starting node \( s \), let 

\[
\text{best}[u] = \text{minimum number of edges in a shortest path from } s \text{ to } u.
\]

In the example below, the best values for nodes \( s, a, b, c, d \) are 0, 1, 2, 1, 2 respectively.

Given an undirected graph \( G = (V, E) \), positive edge weights \( w(e) \) for each edge \( e \) and a starting node \( s \),

a. Design an efficient algorithm that computes the values of \( \text{best}[u] \) for all nodes \( u \in V \).

b. Briefly argue why it is a correct algorithm.

c. What is its runtime?