Question 1. Given $n$ coins, some of which are heavier, find the number of heavy coins using $O(\log^2 n)$ weighings. Note that all heavy coins have the same weight as do all the light ones.

Question 2. Given a permutation of numbers 1 to $n$, count the number of minimal inversion pairs. An inversion pair $(j, i)$ is said to be minimal iff there is no number $k$, $i < k < j$ such that $(j, k)$ and $(k, i)$ are also inversion pairs. In other words the pair $(j, i)$ is not implied by other pairs by transitivity. Use divide and conquer.

Question 3. Given a set of $n$ horizontal and vertical line segments, count the number of intersections between a horizontal and a vertical segment. Assume that all horizontal segments are disjoint as are the vertical segments. Use divide and conquer.

Question 4. Given a sequence of positive numbers $a_1, a_2, \ldots, a_n$ design a linear time algorithm to find a shortest subsequence of consecutive numbers $a_i, a_{i+1}, \ldots, a_j$ whose sum is at least a given number $M$.

Suppose each number is assigned a weight $w_i$, which may be positive or negative. Design a linear time algorithm to find a minimum weight subsequence of consecutive numbers whose sum is at least $M$. (The weight of a subsequence is the sum of weights of the numbers in the subsequence).

Now suppose the numbers $a_i$ can be positive or negative. Do the same as above, the time complexity being $O(n \log n)$ instead of linear.

Question 5. Given a tree $T$ and a set $C$ of subtrees of $T$. Design a greedy algorithm to find a largest subset of $C$ such that all trees in the subset are vertex disjoint.

Question 6. Given a permutation of the numbers 1 to $n$, find a pair of numbers $(i, j)$ with the property that the number of numbers with value between $i$ and $j$ which occur between $i$ and $j$ in the permutation is a maximum. Running time should be anything better than $O(n^2)$. First try getting $O(n^2)$.

Question 7. Let $G$ be a graph with weights assigned to the edges and $R$ be a specified subset of edges of $G$ not containing a cycle. Describe a greedy algorithm to find a minimum weight spanning tree of $G$ containing all edges in $R$. Prove the correctness of your algorithm.

Question 8. An edge of a graph is said to be critical if the weight of a minimum weight spanning tree in the graph is increased if the weight of the edge is increased. Describe an algorithm to find all critical edges in a graph.
Question 9.

1. Let \( G \) be an arbitrary graph and \( R \) a subset of vertices in \( G \). Design a greedy algorithm to find the minimum number of edges to be deleted from \( G \) so that in the resulting graph there is no cycle passing through a vertex in \( R \).

2. What happens if the edges are weighted and it is required to find a minimum weight set of edges instead of minimum number of edges?

Question 10. Assuming that only equality checks are allowed, design an \( O(n) \) time algorithm to test if there is an element which occurs more than \( n/2 \) times in an array containing \( n \) elements.

Acknowledgement: My source for these problems is Prof. Ajit Diwan, IIT Bombay.