All for solutions, provide explanation first in English followed by pseudo code. A brief complexity analysis, including how to derive the result, is also needed.

1. (divide-and-conquer using transform-and-conquer) Compute a mode, where mode is a value that occurs most often in a given list of numbers. For example, for 2, 4, 6, 2, 6, 1, 6, the mode is 6. Design an algorithm with complexity $\Theta(n \log n)$.

2. (divide-and-conquer and dynamic programming) Suppose your job at an investment company is to buy $x$ shares of a stock on some day and sell all these shares on some (later) day. There are $i = 1, 2, \ldots, n$ days. The share price at day $i$ is $p(i)$. Design two efficient algorithms that generate the maximum profit by deciding when to buy and sell. The first solution uses divide-and-conquer with complexity $\Theta(n \log n)$. The second solution applies dynamic programming to reduce the complexity to $\Theta(n)$.


4. (stable marriage problem) Suppose $2n$ people ($n$ men and $n$ women) are either bad or good. Specifically, there are $k$ good men and $k$ good women. In the preference list, everyone would rather marry any good person than any bad person. Show that in a stable marriage, every good man is married to a good woman.

5. (bonus problem) Suppose you are given an array $A$ with $n$ entries with distinct values. Assume the values in the array is unimodal: For some index $p$ between 1 and $n$, the values in the array entries increase up to position $p$ and then decrease the remainder of the way until position $n$. Show how to find the entry $p$ by reading at most $O(\log n)$ entries in $A$ and then how to minimize the total number of readings.