1. A triomino is an L-shaped tile formed by three adjacent squares of a chess board. The problem is to cover any $2^n$ by $2^n$ chess board with one missing square (anywhere on the board) with triominos. Triominos should cover all the squares except the missing one with no overlaps. Design a divide and conquer algorithm for the problem. Prove the worst case running time of your algorithm.

![Triomino](https://via.placeholder.com/150)

2. You are given a pile of $n$ identically looking coins, one of which is fake. The fake coin weighs less than the real coins. Design an efficient algorithm to determine which coin is fake. Prove the worst case running time of your algorithm.

3. There are $n$ pancakes, all of different sizes, that are stacked on top of each other. You are allowed to slip a flipper under one of the pancakes and flip over the whole stack above the flipper. The purpose is to arrange the pancakes according to their size with the biggest at the bottom. Design an efficient algorithm for solving this puzzle. Prove the worst case running time of your algorithm.

4. Will the “median-of-medians” algorithm for the selection problem still work in linear time if the array is divided into groups of 7? (Prove one way or the other.) Prove that the algorithm will not work in linear time if groups of 3 are used.