Weekly Assignment 8: Divide and Conquer

1. We're about to analyse the complexity of a number of divide-and-conquer algorithms. Unfortunately, the algorithms themselves got lost. But fortunately, the recurrences measuring their worst case number of operations are still there! For each item, give the least $\mathcal{O}$-class it belongs to. No proof or explanation needed.

   (a)
$$T(n) = \begin{cases} 1 & \text{if } n = 0 \\ 3T(n-1) & \text{if } n > 0 \end{cases}$$

   (b)
$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2T(\frac{n}{2}) & \text{if } n = 2^k \text{ for some } k > 0 \end{cases}$$

   (c)
$$T(n) = \begin{cases} 37 & \text{if } n = 1 \\ 3T(\frac{n}{3}) + 42 & \text{if } n = 3^k \text{ for some } k > 0 \end{cases}$$

   (d)
$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(\frac{n}{4}) + n & \text{if } n = 4^k \text{ for some } k > 0 \end{cases}$$

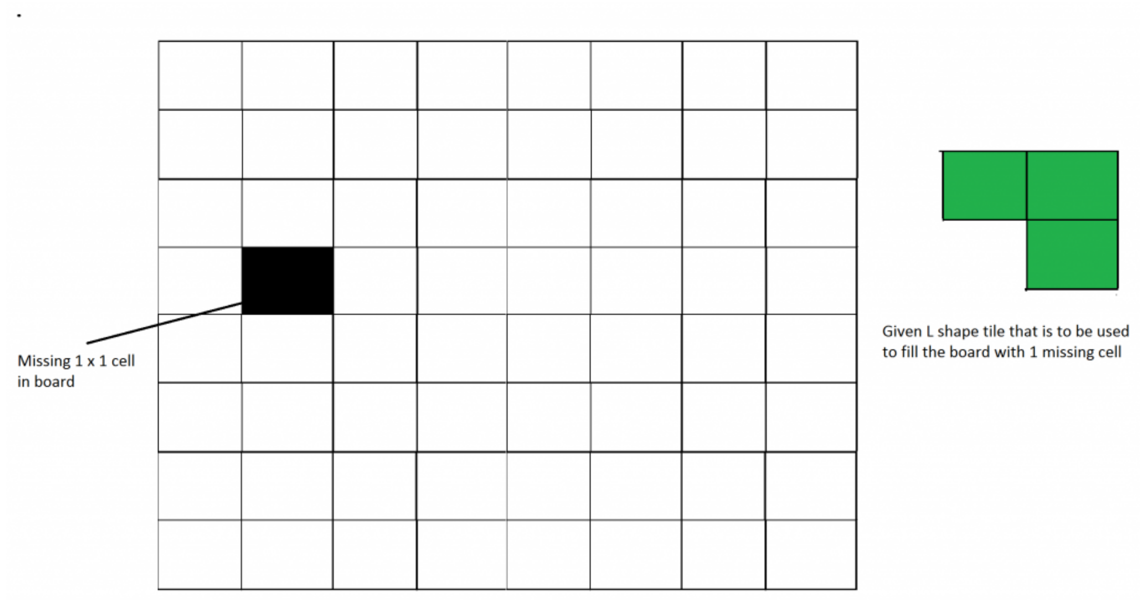
2. The power in a large flat is down, due to short circuit somewhere. You are in charge of identifying the room which causes the problem (there is exactly one room which causes short circuit). You're allowed to do two things:

   - control the main power switch of the building;
   - control the switches of each individual room, but for safety reasons *you can only flip individual switches if the main power switch is off.*

   A room has electricity if and only if both the main switch and the switch for that room are enabled. If the room which short circuits is powered, the main switch will turn off again and you hear a loud "bang", but fortunately nothing will really be broken afterwards.

   Design an algorithm which finds the room with the short circuit problem, where the number of times the main power switch is flipped is in $\mathcal{O}(\lg n)$ (the number of times you flip switches of individual rooms doesn't matter).

3. Let us consider an extremely simplified version of Tetris, with a unique simplified L-shaped tile and a board that consists of $2^n \times 2^n$ cells, for some positive integer n. The game is to place as many tiles (the shape is given below, they can be rotated) as possible on the board. We suppose that there is exactly 1 designated cell on the board that is "missing" and that cannot be covered.


Missing 1 x 1 cell in board
Given L shape tile that is to be used to fill the board with 1 missing cell

Describe and analyze (i.e give a correctness proof and time complexity) a divide-and-conquer algorithm to fully cover the board (except for the missing cell) with L-tiles. The input of the algorithm consists of the value for n and the coordinates of the missing cell, and the output is a list of tiles, where each tile is represented by the three cells on the board that it covers. For instance, for input $(1,(2,2))$ the corresponding output is $(\{(1,1),(1,2),(2,1)\})$.

4. Consider the following sorting algorithm which takes as input a list A of integers. The algorithm first selects the largest element from A, and swaps this with the last element of the array. Then, it recursively sorts the rest.

   During the algorithm, we keep a separate list B of integers that is already sorted, initially empty. The algorithm repeatedly selects the largest element of A, puts it in front of B, and removes it from A. Once A is empty, we return the list B.

   Show that this algorithm is in $\mathcal{O}(n^2)$, by writing down a recurrence for the number of comparisons in the worst case, solving it (by making an educated guess, or by repeated substitution as shown in the lecture) and proving by induction that your solution is correct.

5. Let $A = [a_1, a_2, \ldots, a_n]$ be an integer array of length n. An *inversion* is a pair of indices i, j with $i < j$ such that $a_i > a_j$. For instance, in the (1-indexed) array

   $$[5, 4, 6, 2]$$

   there are 4 inversions: $(1,2), (1,4), (2,4)$ and $(3,4)$.

   Describe an efficient algorithm based on divide and conquer, which takes an integer array A as input, and returns the number of inversions in A. Explain your algorithm, and its correctness and runtime complexity.