Problem 1 (Divide and Conquer the Peak, 30 pts)

Suppose we have an array A of n distinct integers and moreover are guaranteed that the array has the following property: up to some index $1 \leq i \leq n$, A is increasing, i.e., $A[1] < A[2] < \ldots < A[i]$, and then after index i, A is decreasing, i.e., $A[i] > A[i+1] > \ldots > A[n]$. In this array, we call $A[i]$ the *peak* of A. For example, consider the array $[1, 4, 7, 8, 6, 2]$, which has peak 8.

Create an algorithm that finds the peak of an input array A in sublinear time (i.e. $o(n)$). Justify the correctness and time complexity of your proposed algorithm.

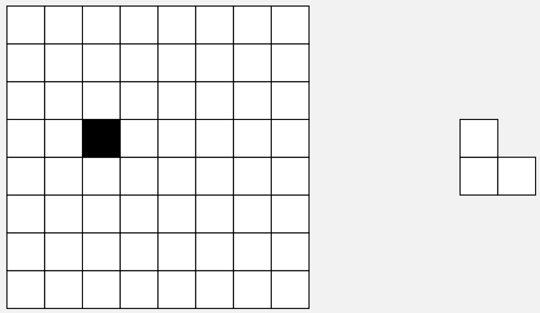
Problem 2 (Two-Server Search, 35 pts)

You have (limited) access to two databases, each of which contains n numbers. For simplicity, you may assume each value across the $2n$ entries is unique. You may only query the databases in the following way: You give one of the databases an integer $1 \leq k \leq n$, and it responds with the k-th smallest number in its database.

Give a divide-and-conquer algorithm that finds the median value across all $2n$ entries using $O(\log n)$ queries to the servers. Justify the correctness and time complexity of your proposed algorithm.

Problem 3 (Grid Tiling, 35 pts)

For a positive integer n, consider a $2^n \times 2^n$ grid, where one of the unit squares is black and all others are white. Show that regardless of the position of the black unit square, the white area can be fully covered, without any overlapping, by L-shaped tiles consisting of 3 unit squares (rotations are allowed).



Left: an example grid with $n = 3$; Right: the L-shaped tile

(*Hint: Use a proof by construction, i.e. design a divide-and-conquer algorithm that takes as input n and (i, j) with $1 \leq i, j \leq 2^n$, and outputs a way to tile a $2^n \times 2^n$ grid with the black square at position (i, j).*)