This assignment is about recursion and solving by divide and conquer. For all the following problems, give recursive solution (even if better iterative solution exists), prove correctness by induction, solve run time and memory usage by recurrence.

1. **CLRS: Problems 2-1: Insertion sort on small arrays in merge sort**

2. **CLRS: Problems 3-2: Relative asymptotic growths**

3. **CLRS (Page 85): Problems 4-2: Finding the missing integer**
   
   An array $A[1..n]$ contains all the integers from 0 to $n$ except one. It would be easy to determine the missing integer in $O(n)$ time by using an auxiliary array $B[0..n]$ to record which numbers appear in $A$. In this problem, however, we cannot access an entire integer in $A$ with a single operation. The elements of $A$ are represented in binary, and the only operation we can use to access them is "fetch the $j$-th bit of $A[i]"," which takes constant time.

   Show that if we use only this operation, we can still determine the missing integer in $O(n)$ time.

4. **Tiling with L-shaped tiles**:
   
   We are given a square chess-like board of size $2^k$ by $2^k$. One of $2^k$ unit-squares in the board is broken. We are required to cover the remaining part by L-shaped tiles of size 3. An L-shaped tile is obtained by removing any of the 4 unit-squares from a 2 by 2 square tile.

   Try to reduce this problem to 4 sub-problems. The division into sub-problem is similar to the division of a square matrix in 4 parts (as we saw in Strassen’s algorithm).

5. **Maximum Subsequence sum**:
   
   We have an array (or sequence) of integers (not necessarily positive). We are required to find a contiguous sub-array (or subsequence) for which the sum of elements in it are maximized.
Try to reduce this problem to 2 sub-problems (like in merge-sort). The subsequence you are looking for are either fully contained in the sub-problems or is made of some suffix of one part and some prefix of the other part.