Problem 1  For each of the following recurrences, use the master theorem to give the tight asymptotic upper bound. You just need to give the final bound for each recurrence.

(a) $T(n) = 4T(n/3) + O(n)$.  
$b(n) = O(\_\_\_\_\_\_\_\_\_\)$.  
(b) $T(n) = 3T(n/3) + O(n^2)$.  
$b(n) = O(\_\_\_\_\_\_\_\_\_\)$.  
(c) $T(n) = 4T(n/2) + O(n^2\sqrt{n})$.  
$b(n) = O(\_\_\_\_\_\_\_\_\_\_\)$.  
(d) $T(n) = 8T(n/2) + O(n^3)$.  
$b(n) = O(\_\_\_\_\_\_\_\_\_\_\)$.  

Problem 2  We consider the following problem of counting stronger inversions. Given an array $A$ of $n$ positive integers, a pair $i, j \in \{1, 2, 3, \cdots, n\}$ of indices is called a strong inversion if $i < j$ and $A[i] > 2A[j]$. The goal of the problem is to count the number of strong inversions for a given array $A$. Give a divide-and-conquer algorithm that runs in $O(n \log n)$ time to solve the problem.

Problem 3  Given an array $A$ of $n$ distinct numbers, we say that some index $i \in \{1, 2, 3, \cdots, n\}$ is a local minimum of $A$, if $A[i] < A[i-1]$ and $A[i] < A[i+1]$ (we assume that $A[0] = A[n+1] = \infty$). Suppose the array $A$ is already stored in memory. Give an $O(\log n)$-time algorithm to find a local minimum of $A$.

Problem 4  Consider a $2^n \times 2^n$ chessboard with one arbitrary chosen square removed. Prove that any such chessboard can be tiled without gaps by L-shaped pieces, each composed of 3 squares. Figure 1 shows how to tile a $4 \times 4$ chessboard with the square on the left-top corner removed, using 5 L-shaped pieces. Use divide-and-conquer to solve the problem.