1. Solve the recurrence $T(n) = T(\sqrt{n}) + 1$. Give a θ bound.

2. Consider the following functions: $n - \log n, \log\log n, 2^{\log^2 n}, \sqrt{n/\log n}, n^{\log n}$. Arrange them in order s.t. the ith function is $O()$ of the $i+1$th function.

   You may want to note that if $\lim_{n\to\infty} \frac{f(n)}{g(n)} < c$ for any constant c, then $f(n) = O(g(n))$. It is fine if you use this.

3. Consider an array $x[1..n]$. An element $x[i]$ is a local minimum if it is no larger than $x[i-1]$ (if any), as well as $x[i+1]$ (if any). Give an $O(\log n)$ algorithm to find a local minimum.

4. The input to this problem are two sorted arrays, of length m, n respectively. All the $m + n$ values in the two arrays are distinct. Another input is an integer s, satisfying $s \le m + n$. The goal is to find the sth smallest value from the $m+n$ values. Show that this can be done using at most $1 + \lceil \log s \rceil$ comparisons.