Note: Be concise and precise. Just because you write a lot does not mean you get more points. I will take off points if you have irrelevant information.

1) An array of positive or negative integers of size n, where \( A[1] < A[2] < \ldots < A[n] \). Write an algorithm to find \( A[m] = m \) provides such an \( m \) exits. What is the order of execution of your algorithm? Prove a lower bound of \( \log(n) \).

2) Prove that \( 2^n - 1 \) moves are necessary and sufficient for the towers of Hanoi problem.

3) We can improve the running time of quicksort by taking advantage of the fast running time of insertion sort when its input is “nearly” sorted. Upon calling quicksort on a subarray with fewer than \( k \) elements, let it simply return without sorting the subarray. After the top-level call to quicksort returns, run insertion sort on the entire array to finish the sorting process. Argue that this sorting algorithm runs in \( O(nk + n \log(n/k)) \) expected time. How should we pick \( k \), both in theory and in practice?

4) Suppose you have a “black box” worst-case linear time median finding subroutine, give a simple linear-time algorithm that solves the selection problem for an arbitrary order statistic.