

Theory Problems 3

Complexity and Sorting

Problem 1: Big-O

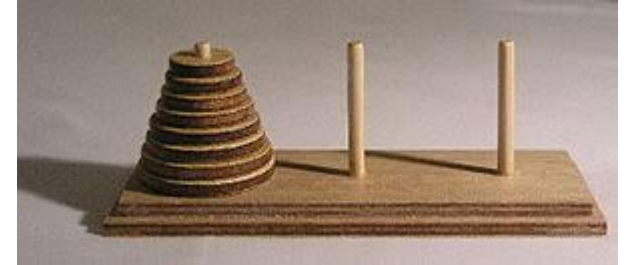
Give the complexity order of the following functions, in terms of the simplest function representing the order, e.g. $2n + 4 \in O(n)$

1. $f(n) = 7n^2 + 800n + 10$
2. $f(n) = \log 2n + 1000000$
3. $f(n) = n! + e^{5n}$
4. $f(n) = 5n \log n + 10 \log n$
5. $f(n) = n \log n^2 + n^2 \log n$
6. $f(n) = e^{3n}$

Problem 2: Tower of Hanoi

The Tower of Hanoi is a puzzle (right) where the objective is to move all the discs from one peg to another, by moving the discs one at a time to another peg. You are not allowed to place a larger disc on a smaller disc at any point.

1. Write an algorithm to move the discs, for any number of discs. The problem is most easily solved by recursion.
2. Prove that your algorithm is correct.
3. What is the complexity of your algorithm?



Problem 3: Integer Sort

You need to sort a list of non-negative integers ($x_i \in \mathbb{N}, i = 1 \dots n$) into order. Consider the following strategy

1. Find the maximum in the list, x_m
2. Create an array of size x_m
3. Go through the list and mark the array at position x_i for each number in the list
4. Run through the array to output the numbers in order

Example (2,4,1) – we create an array of size 5 and mark positions 2,4 and 1. Running through the array in order, we encounter 1,2,4 marked in the array and output in the correct order.

Write a complete algorithm for this sort, also considering the possibility that the same number may be repeated multiple times in the input list. What is the complexity of your algorithm?