University of Zurich[UZH]

Department of Informatics
Database Technology Group
Prof. Dr. M. Böhlen

Informatics II
Exercise 4

March 11, 2019

Divide and Conquer

The three way max finder algorithm takes as input an array of integers and divides it into three equal partitions: **I** (A[0] ... A[*mid1*-1]), **II** (A[*mid1*] ... A[*mid2*-1]) and **III** (A[*mid2*] ... A[n-1]). Afterwards, it calls itself recursively on the three partitions until each partition has only one element. It finds the maximum value for each partition and then continue merging the partitions finding the maximum value of the array.

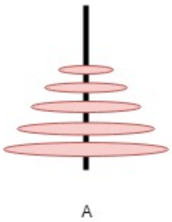
Task 1. Based on the three way max finder description, draw a tree to illustrate the process of finding the array's maximum value, given `A = [24, 7, -18, 9, -7, -6, 15, 2, 1]`. Also write the recurrence relation for this three way max finder algorithm.

Task 2. Provide a pseudocode for three way max finder algorithm. Also implement the algorithm as a C program.

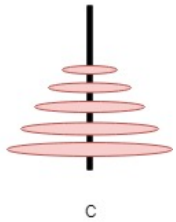
Task 3. The algorithm splits the given array into three partitions. Suppose the algorithm splits the input array into four partitions. What will be the recurrence relation in this case? Would this affect the asymptotic complexity of the algorithm? Explain your answer.

In the Towers of Hanoi problem, there are three sticks and five disks of different sizes. Each disk has a hole through the center so that it fits on a stick. At the start, all five disks are on stick A as shown below. The disks are arranged by size so that the smallest is on top and the largest is on the bottom. The goal is to end up with all five disks in the same order, but on a C stick. There are two restrictions. First, the only permitted action is removing the top disk from a stick and placing it onto another stick. Second, a larger disk can never lie above a smaller disk on any stick. Figure 1 shows initial and goal position of disks. Figure 2 shows allowed actions. Figure 3 shows prohibited actions.

Task 4. Derive the recurrence relation for the Towers of Hanoi problem and calculate the asymptotic tight bound.



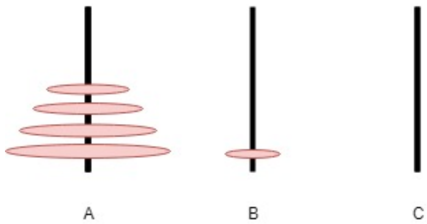
(a) Initial position



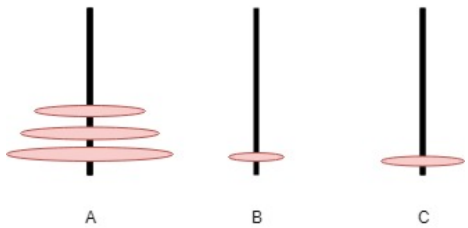
(b) Goal position

Figure 1: Initial and goal position

(a) Step 1



(b) Step 2



(c) Step 3

Figure 2: Allowed actions

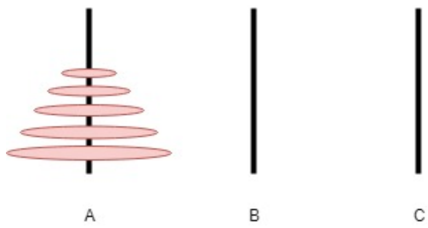
Recurrences

Task 5. Consider the following recurrence:

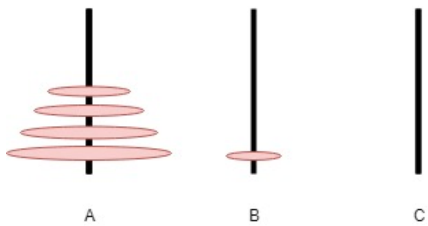
$$T(n) = \begin{cases} 1 & \text{, if } n = 1 \\ T(n/3) + T(n/6) + T(n/9) + n & \text{, if } n > 1 \end{cases}$$

a) Draw a recursion tree and use it to estimate the asymptotic upper bound of $T(n)$. Demonstrate the tree-based computations that led to your estimate.

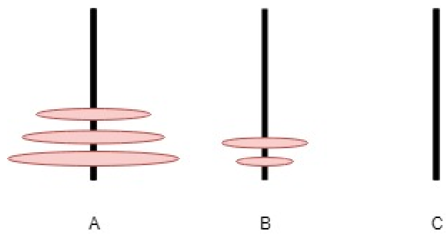
b) Use the substitution method to prove that your estimate in (a) is correct.



(a) Step 1



(b) Step 2



(c) Step 3

Figure 3: Prohibited actions

Task 6. Calculate the asymptotic tight bound of the following recurrences. If the Master Theorem can be used, write down a, b, $f(n)$ and the case (1-3).

1. $T(n) = 16T(\frac{n}{4}) + 6n^2$
2. $T(n) = T(\frac{n}{2}) + 1$
3. $T(n) = 4T(\frac{n}{2}) + 2n$
4. $T(n) = 2T(n-2) + n$
5. $T(n) = 2T(\frac{n}{4}) + n \lg n$