

EL9343 Homework 3

(Due October 4th, 2021)

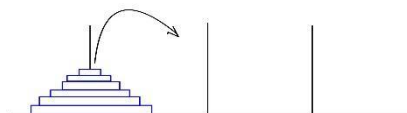
No late assignments accepted

All problem/exercise numbers are for the third edition of CLRS textbook

1. Given the array $A[19, 5, 9, 52, 26, 35, 61, 28]$
 - a) Using Figure 2.2 on page 18 of CLRS as a model, illustrate the operation of insertion-sort on A .
 - b) Using Figure 2.4 on page 35 of CLRS as a model, illustrate the operation of merge-sort on A .
2. Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A , and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A .
 - a) Write pseudocode for this algorithm, which is known as **selection sort**.
 - b) What loop invariant does this algorithm maintain?
 - c) Why does it need to run for only the first $n-1$ elements, rather than for all n elements?
 - d) Give the best-case and worst-case running times of selection sort in Θ -notation.
3. For the maximum subarray problem, if we use divide-conquer, but instead of dividing the array into two halves, we equally divide it into three segments, how should the algorithm be modified? What is the running time of the new algorithm?

Note: Pseudocode is not necessary. You can explain your algorithm and running time step by step.

4.



The [Tower of Hanoi](#) is a mathematical game or puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with n disks stacked on a **start** rod in order of decreasing size, the smallest at the top, thus approximating a conical shape. The objective of the puzzle is to move the entire stack to the **end** rod, obeying the following rules:

1. Only one disk may be moved at a time.
2. Each move consists of taking the top disk from one of the rods and placing it on top of another rod or on an empty rod.
3. No disk may be placed on top of a disk that is smaller than it.

To print each move of this game with the minimum number of steps, please fill in the pseudo code in the *MOVE* function.

For example, ideal output format should be like this:

>>> *MOVE(3, 1, 3)*

Move the top disk from rod 1 to rod 3

Move the top disk from rod 1 to rod 2

Move the top disk from rod 3 to rod 2

Move the top disk from rod 1 to rod 3

Move the top disk from rod 2 to rod 1

Move the top disk from rod 2 to rod 3

Move the top disk from rod 1 to rod 3

Your Answer:

PRINT(origin, destination):

print("Move the top disk from rod", origin, "to rod", destination)

MOVE(n, start, end):

your code

your code

your code

Hint: You can try Divide and Conquer or Recursion. You can use the following code form as inspiration.

MOVE(n, start, end):

if n == 1:

else:

 MOVE(____,____,____)

 MOVE(____,____,____)

 MOVE(____,____,____)

5. For an array with n elements, design a *divide-and-conquer* algorithm for finding both the minimum and the maximum element of this array using no more than $3n/2$ comparisons. (Pseudocode is required)