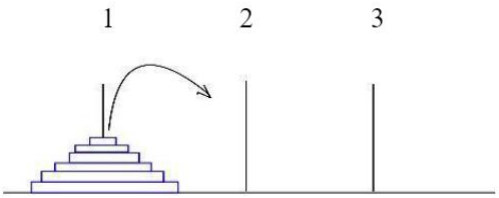
EL9343 Homework 3

Due: Sep.29th 11:00 a.m.

1. True or False questions:
   (a) One can find the maximum sub-array of an array with n elements within $O(n \log n)$ time.
   (b) In the maximum sub-array problem, combining solutions to the sub-problems is more complex than dividing the problem into sub-problems.
   (c) Bubble sort is stable.
   (d) When input size is very large, a divide-and-conquer algorithm is always faster than an iterative algorithm that solves the same problem.
   (e) It takes $O(n)$ time to check if an array of length n is sorted or not.
   (f) Insertion sort is NOT in-place.
   (g) The running time of merge-sort in worst-case is $O(n \log n)$.
2. Consider sorting n numbers stored in array A, indexed from 1 to n. First find 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 pseudo-code for this algorithm, which is known as *selection sort*.
   (b) What loop invariant does this algorithm maintain?
   (c) Give the best-case and worst-case running times of selection sort in Θ-notation.
3. A mathematical game (or puzzle) consists 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:
   i Only one disk may be moved at a time;
   ii 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;
   iii No disk may be placed on top of a disk that is smaller than it.



Please design a MOVE(n, start, end) function to illustrate the minimum number of steps of moving n disks from start rod to the end rod.
You **MUST** use the following functions and format, otherwise you will not get points of part (a) and (b):

```
def PRINT(origin, destination):
        print("Move the top disk from rod", origin, "to rod", destination)

def MOVE(n, start, end): # TODO: you need to design this function
        pass
```

For example, the output of MOVE(2, 1, 3) should be:

```
Move the top disk from rod 1 to rod 2
Move the top disk from rod 1 to rod 3
Move the top disk from rod 2 to rod 3
```

(a) Give the output of MOVE(4, 1, 3).

(b) Fill in the function MOVE(n, start, end) shown above. You can use Python, C/C++ or pseudo-code form, as you want.

(c) What's the minimum number of moves of MOVE(5, 1, 3), and MOVE(n, 1, 3)?

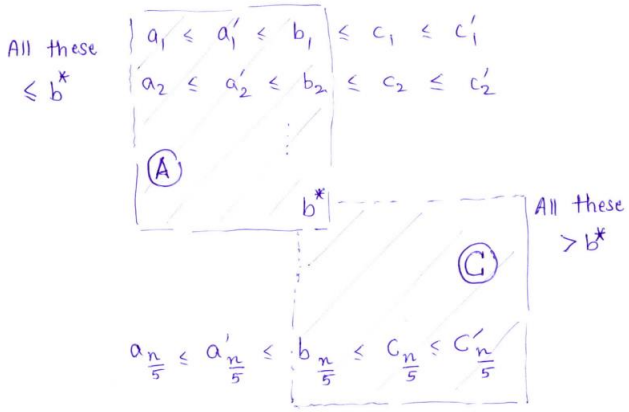
4. Finding the median of an unordered array in $O(n)$ (Part I). Let's consider the algorithm 1,

---

Algorithm 1 FindMedian(L)

Input: L as input list containing n real numbers
Output: The median of L as m
if $n \leq 10$ **then**
Sort L and return the median m
end if

Divide L into $\frac{n}{5}$ lists of size 5 each
Sort each list, let i^{th} list be $a_i \leq a_i' \leq b_i \leq c_i \leq c_i', i = 1, 2, \ldots, \frac{n}{5}$
Recursively find median of $b_1, b_2, \ldots, b_{\frac{n}{5}}$, call it b^*
Reorder indices so that $b_1, b_2, \ldots, b_{\frac{n}{10}} \leq b^* < b_{\frac{n}{10}+1}, \ldots, b_{\frac{n}{5}}$
Define A and C as shown in the figure (both A and C have approximately $\frac{3}{10}n$ elements)



Drop A and C from the original list L, to get a new list L', with $n - \frac{3}{10}n - \frac{3}{10}n = \frac{2}{5}n$ elements
Find median of remaining L' recursively and return it as m.

Note that there could be some corner cases to consider, like n is not a multiple of 5 (can be fixed by padding), or the number of elements in A or C is inaccurate. However, as long as n is large enough, this little inaccuracy will not infect the analysis of complexity.

(a) Let $T(n)$ be the running time of this algorithm, find the recurrence formula, and then solve it.

(b) Is this algorithm correct? If yes, try to prove it; otherwise, find a counter case (like the true median is in A or C and is dropped).