German University in Cairo
Department of Computer Science
Dr. Nourhan Ehab

CSEN 703 Analysis and Design of Algorithms, Winter Term 2024
Practice Assignment 1

Exercise 1-1 From CLRS (©MIT Press 2001)

Illustrate the operation of Insertion Sort on the array $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

Exercise 1-2

Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion on A.

i. List the five inversions of the array $(2, 3, 8, 6, 1)$. Note that inversions are specified by indices rather than by the values in the array.

ii. Identify the array containing all the elements from the set $\{1, 2, ..., n\}$ which has the most inversions. How many does it have?

iii. Is there a relationship between the operations performed in insertion sort given an input array, and the number of inversions?

Exercise 1-3 From CLRS (©MIT Press 2001)

Consider the **searching problem**:
Input: A sequence of n numbers $A[a_1, a_2, ..., a_n]$ and a value v.
Output: An index i such that $v = A[i]$, or the special value NIL if v does not appear in A.

i. Write pseudocode for linear search which scans through the sequence looking for v.

ii. Analyze the best and worst-time complexity for linear search.

iii. Using a loop invariant, prove that your algorithm is correct.

Exercise 1-4

Write an algorithm to find the index of the largest and smallest value in an array of integers.

i. What is the best-case and worst-case running times of your algorithm?

ii. Define the loop invariant for your algorithm and show that it holds.

Exercise 1-5

The following code snippet computes the sum of the first n numbers in the array a.

```
sum ← 0
```

```
for i = 1; i ≤ A.length; i++ do
    sum ← sum + A[i]
end for
```

i. What is the best-case and worst-case running time of the above algorithm?

ii. What is the loop invariant?

iii. Prove that your invariant holds.

Exercise 1-6

Consider sorting n numbers in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then, finding the second smallest element of A, and exchanging it with $A[2]$. Continue in this manner for the first $n-1$ elements of A. Write pseudo code for this algorithm, which is known as **selection sort**.

i. Give the best-case and worst-case running times of selection sort.

ii. Why does the algorithm need to run for only the first $n-1$ elements, rather than for all n elements?

iii. Prove that selection sort is correct.

Exercise 1-7

Consider the following pseudo code for the algorithm Gnome Sort:

```
function GnomeSort(Array A)
    i ← 1
    while i ≤ n do
        if (i == 1) or (A[i − 1] ≤ A[i]) then
            i++
        else
            Exchange A[i] ↔ A[i − 1]
            i−−
        end if
    end while
end function
```

i. Provide an example for each of best and worst-case inputs.

ii. Analyze the best and worst-time complexity of gnome sort.

iii. Prove that the algorithm is correct.