DATA STRUCTURES AND ALGORITHMS
ASSIGNMENT 2

AAKASH JOG
ID : 989323563

Exercise 1.
The following is a pseudocode of the Bubble Sort Algorithm.

Input: $A[1:n]$, n

```
1: procedure BUBBLE SORT(A[1 : n], n)
2:     for i = n − 1 to 1 do
3:         for j = 1 to i do
4:             if A[j] > A[j + 1] then
5:                 SWAP(A, j, j + 1)¿
6:             end if
7:         end for
8:     end for
9: end procedure

10: procedure SWAP(A, j, k)
11:     x ← A[j]
12:     A[j] ← A[k]
13:     A[k] ← x
14: end procedure
```

---

(1) Which property does the array hold at the end of the ith iteration of the for loop?
(2) What is the algorithm's running time?

Solution 1.
(1) At the end of the ith iteration of the second for loop, the sub array $A[1:i]$ is sorted in ascending order.
(2) As the algorithm has one for loop inside another, with each of the iterable changing in a linear manner, the running time of the algorithm is $\mathrm{O}\left(n^2\right)$.

Exercise 2.
We've seen in class that the running time of Insertion Sort on different inputs is different. In particular, when the array is sorted in descending order (from the largest to the smallest) the running time is quadratic, while when the array is sorted in increasing order (from the smallest to the largest) the running time is linear. In this question, we try to understand more generally,

Date: Tuesday 29th March, 2016.

how the running time for different inputs is dependent on "how much they are not sorted".
Given an array A of length n. Let the running time of Insertion Sort of A be denoted by $T_{\mathrm{IS}}(A)$. For each index $1 \leq j \leq n$, let the number of elements in A whose value is larger than $A[j]$ but appear before $A[j]$ be denoted by $v_A(j)$.

(1) Give an example of an array A of length $n = 5$ that is sorted in ascending order. For each index $1 \leq j \leq 5$, what is the value of $v_A(j)$?
(2) Give an example of an array A of length $n = 5$ that is sorted in descending order. For each index $1 \leq j \leq 5$, what is the value of $v_A(j)$?
(3) What is the relation between $v_A(j)$ and the number of iterations of the while loop on array A in the jth iteration of the for loop?
(4) What is $T_{\mathrm{IS}}(A)$ in terms of Θ as a function of $v_A(1), \ldots, v_A(n)$? Explain your answer.

Solution 2.
(1)

$$A = [1, 2, 3, 4, 5]$$

Therefore,

$$v_A(1) = 0$$
$$v_A(2) = 0$$
$$v_A(3) = 0$$
$$v_A(4) = 0$$
$$v_A(5) = 0$$

(2)

$$A = [5, 4, 3, 2, 1]$$

Therefore,

$$v_A(1) = 0$$
$$v_A(2) = 1$$
$$v_A(3) = 2$$
$$v_A(4) = 3$$
$$v_A(5) = 4$$

(3) $v_A(j)$ is equal to the number of iterations of the while loop in the jth iteration of the for loop. This is because $v_A(j)$ represents the number of elements of A to the left of $A[j]$ which need to be moved to the right of $A[j]$.
(4) $v_A(j)$ is equal to the number of iterations of the while loop in the jth iteration of the for loop. Therefore, the number of operations in the jth iteration of th for loop is $c_1 + c_2 v_A(j)$. Therefore, the total running time

is

$$T_{\text{IS}}(A) = \sum_{j=2}^{n} c_1 + c_2 v_A(j)$$

where each $v_A(j)$ is dependent on n and the original sorting of the array.

Exercise 3.
Consider the following algorithm. Let the running time of the three proce-

```
1: procedure ALG(n)
2:     PROC1(n)
3:     PROC2(n)
4:     PROC3(n)
5: end procedure
```

dures PROC1, PROC2, and PROC3 be $T_1(n)$, $T_2(n)$, and $T_3(n)$ respectively. Let the running time of the entire algorithm be $T_A(n)$.
Let

$$T_1(n) = \Omega\left(n \log(n)^3\right)$$
$$T_2(n) = \mathrm{O}\left(n^2\right)$$
$$T_3(n) = \Theta\left(n^{0.5}\right)$$

Which of the following bounds are necessarily true, which are necessarily not true, and which are not necessarily true? Explain your answer.

(1) $T_A(n) = \Omega\left(n^{0.5}\right)$
(2) $T_A(n) = \Omega\left(n^2\right)$
(3) $T_A(n) = \mathrm{O}(n)$

Solution 3.
(1) In the asymptotic case, i.e. for $n \to \infty$,

$$n^{0.5} < n \log(n)^3 < n^2$$

Also,

$$T_2(n) = \mathrm{O}\left(n^2\right)$$

Therefore,

$$T_A(n) = \Omega\left(n^{0.5}\right)$$

is necessarily true.
(2) In the asymptotic case, i.e. for $n \to \infty$,

$$n^{0.5} < n \log(n)^3 < n^2$$

Also,

$$T_2(n) = \mathrm{O}\left(n^2\right)$$

Therefore,

$$T_A(n) = \Omega\left(n^2\right)$$

is not necessarily true.

(3) In the asymptotic case, i.e. for $n \to \infty$,

$$n^{0.5} < n\log(n)^3 < n^2$$

Also,

$$T_1(n) = \Omega\left(n\log(n)^3\right)$$

Therefore,

$$T_A(n) = \mathrm{O}(n)$$

is necessarily not true.