The National School of Artificial Intelligence
المدرسة الوطنية العليا للذكاء الاصطناعي

Data Structures & Algorithms 2
Homework #1

Submission deadline : Thursday 09/11/2023

Note: You are required to submit your homework in the form of (**pdf**) file on your group's classroom.

Exercise 1

For each of the functions f(N) given below, indicate the tightest bound possible (in other words, giving $O(2^N)$ as the answer to every question is not likely to result in many points). Unless otherwise specified, all logs are base 2. Give better insights of how you solved this?

a) $f(N) = 100 \log N^2 + 10 N^2 \log N$

b) $f(N) = ((N + 1)(N + 2))/2$

c) $f(N) = N^2 (2\log N + \log N) + N^3$

d) $f(N) = N \log^2 N + N \log \log N$

e) $f(N) = N^2 (N + 2N) + (N^3 . N^3)$

f) $f(N) = N^{1/4} + \log N$

Exercise 2

Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable n. Justify your answer.

| **(A)** | ```
void fct1(int n)  {
    for (int i = n*n; i > 0; i--) {
        for (int k = 0; k < n; ++k)
            print("k = " , k);
        for (int j = 0; j < i; ++j)
            print("j = " , j);
        for (int m = 0; m < 5000; ++m)
            print("m = " , m);
    }
}
``` |

| **(B)** | ```
int fct2 (int n,  int m) {
      if (n < 10) return n;
      else if (n < 100)
              return fct2  (n - 2, m);
         else
              return fct2 (n/2, m);
}
``` |
| **(C)** | ```
void fct3 (int n) {
      for (int i = 0; i < n; ++i) {
         for (int j = 0; j < n; ++j)
            print("j = "  j);
         for (int k = 0; k < i; ++k) {
            print("k = " ,, k);
             for (int m = 0; m < 100; ++m)
                 print("m = ", m);
         }
      }
}
``` |

Exercise 3

Suppose you have a large linked list of n integers and you want to print them in reverse order (the numbers closer to the end of the list first).

The first version of your code follows this algorithm:

- Traverse the list from the beginning to determine what n is.
- For i = n, n − 1, n − 2, ..., 1, traverse the list from the beginning to the i th element and print it.

The second version of your code looks like this, calling ***printReverse*** on the first node in the list.

```
class ListNode {

      int x;

      ListNode next;

      Public

      void printReverse() {

            if (next != null) next.printReverse();

            print(x);

      }

 }
```

- Give an asymptotic analysis of the running time using big-O for both algorithms .Which version is faster?