The National School of Artificial Intelligence
المدرسة الوطنية العليا للذكاء الاصطناعي

2^{nd} Year /Semester3
(2024-2025)

Data Structures & Algorithms 2
Tutorial 2
Algorithm Analysis -part 2

OBJECTIVE

Compute the computational complexity for a recursive algorithm

Exercise 1

Using the standard master theorem, give asymptotic upper and lower bounds for T(n) in each of the following recurrences. Assume that T(n) is constant for sufficiently small n. Make your bounds as tight as possible, and justify your answers.

A. $T(n) = 3T(n/2) + n^2$

B. $T(n) = 4\,T(n/2) + n^2\sqrt{n}$

C. $T(n) = \sqrt{2}\,T(n/2) + \log n$

D. $T(n) = 2^n T(n/2) + n^n$

E. $T(n) = 3T(n/2) + n \log n$

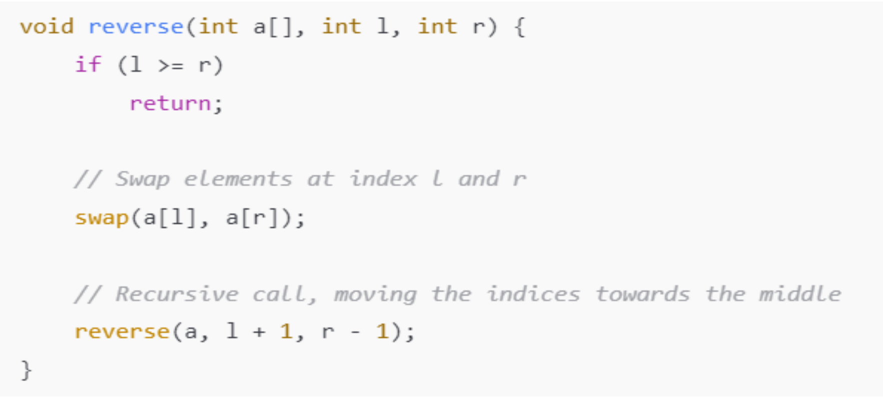
F. $T(n) = 4T(n/2) + n^2$

G. $T(n) = 16T(n/4) + n!$

H. $T(n) = 2T(n/2) + n/\log n$

Exercise 2

Analyse the time complexity of the recursive function ***reverse*** in terms of the input size.



```
void reverse(int a[], int l, int r) {
    if (l >= r)
        return;

    // Swap elements at index l and r
    swap(a[l], a[r]);

    // Recursive call, moving the indices towards the middle
    reverse(a, l + 1, r - 1);
}
```