UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA EP DE INGENIERÍA DE SISTEMAS PERIODO ACAD.2025-1



**LABORATORIO°4**

CURSO: Diseño y Análisis de Algoritmos

DOCENTE: Jorge Luis Chávez Soto

ALUMNO:

RAFAEL IVAN RUIZ MOLINA     21200125

**Análisis orden asintótico (Multiplicación de Polinomios)**

**Estructura general del algoritmo:**

Método principal:

public static int[] multiplicarRecursivo(int[] A, int[] B, int gradoA, int gradoB)

Este método inicializa el arreglo **resultado** y llama al método recursivo **multiplicarTermino**. Esta parte tiene tiempo constante más la llamada recursiva, así que el trabajo pesado está en los métodos recursivos siguientes.

1. Método multiplicarTermino:

private static void multiplicarTermino(int[] A, int[] B, int[] resultado, int i, int gradoA, int gradoB)

Este método itera recursivamente sobre todos los términos de **A**(índice **i**va de **0** a **gradoA**), y por cada término llama a**multiplicarInterno**. Entonces, se ejecuta **(gradoA + 1)** veces.

1. Método multiplicarInterno:

private static void multiplicarInterno(int[] A, int[] B, int[] resultado, int i, int j, int gradoB)

Por cada índice *i*, este método recorre recursivamente todos los términos de **B** (índice **j** va de **0** a **gradoB**), ejecutando **A[i] \* B[j]**. Se ejecuta **(gradoB + 1)** veces por cada llamada de **multiplicarTermino**.

**Función tiempo (T(n)):**

Sea **n = gradoA + 1** y **m = gradoB + 1** (cantidad de términos de A y B respectivamente):

* **multiplicarTermino** se llama **n** veces.
* **multiplicarInterno** se llama **m** veces dentro de cada llamada de **multiplicarTermino**.

Por tanto, el número total de operaciones recursivas es:

***T(n,m)= n ⋅ m***

**Notación asintótica:**

La tasa de crecimiento del algoritmo es:

***O(n ⋅ m)***

donde:

* **n** es el número de términos del polinomio A (**gradoA + 1**)
* **m** es el número de términos del polinomio B (**gradoB + 1**)