

Instituto Politécnico Nacional
Escuela Superior de Cómputo
Análisis de algoritmos
Práctica 1
Alumno: AbrMa

1. Función para multiplicación de matrices

```
int **multiplicacion_matrices(int **matrizA, int **matrizB, int n, int m, int o, int p)
{
    int ** resultado = crea_matriz(n,p);
    int i, j, k, casilla = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < p; j++)
        {
            for (k = 0; k < m; k++)
                casilla += matrizA[i][k] * matrizB[k][j];
            resultado[i][j] = casilla;
            casilla = 0;
        }
    return resultado;
}
```

2. Calculo de complejidad

Solución: Analizando la función anterior podemos escribir la complejidad C como

$$C = 8(1) + \sum_{n=1}^n [5 + \sum_{p=1}^p (8 + \sum_{m=1}^m 8)]$$

$$C = 8(1) + \sum_{n=1}^n [5 + \sum_{p=1}^p (8 + 8\{m - 1 + 1\})]$$

$$C = 8 + \sum_{n=1}^n [5 + \sum_{p=1}^p (8 + 8m)]$$

$$C = 8 + \sum_{n=1}^n [5 + \{(8 + 8m)(p - 1 + 1)\}]$$

$$C = 8 + \sum_{n=1}^n [5 + 8p + 8pm]$$

$$C = 8 + (5 + 8p + 8pm)(n - 1 + 1)$$

$$C = 8 + 5n + 8np + 8npm$$

De lo anterior podemos ver que

$$O(8 + 5n + 8np + 8npm) \approx O(npm) \approx O(n^3)$$