

Programación: producto punto de dos vectores

Objetivos. Programar la función que calcule el producto punto de dos vectores.

Requisitos. Programación de funciones, ciclos `for`, entradas de vectores.

1. Problema: producto punto de dos vectores. En algún lenguaje de programación escribir una función que calcule el producto punto de dos vectores dados.

Entrada: $a, b \in \mathbb{R}^n$.

Condición que debe cumplir la entrada: los vectores dados son de la misma longitud. No tiene que verificar esta condición.

Salida: el número $\sum_{j=1}^n a_j b_j$.

Solución en el lenguaje de Matlab (guardar en el archivo `mydotproduct.m`):

```
function [p] = mydotproduct(a, b),
    n = length(a);
    p = 0;
    for j = 1 : n,
        p = p + a(j) * b(j);
    end
end
```

2. Análisis de complejidad. Calcular el número de operaciones de multiplicación en el algoritmo anterior. La respuesta es un polinomio de la variable n . Calcular el número de operaciones de adición.

3. Pruebas con vectores pequeños. En algún lenguaje de programación escribir un programa que llame a la función del ejercicio anterior, aplicándola a los vectores

$$a = \begin{bmatrix} 3 \\ -4 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 5 \\ -2 \end{bmatrix}.$$

Solución en el lenguaje de Matlab (guardar en el archivo `testmydotproduct1.m`):

```
function [] = testmydotproduct1(),
    a = [3; -4; 1]; b = [2; 5; -2];
    display(mydotproduct(a, b));
    display(a' * b);
end
```

4. Pruebas con vectores pseudoaleatorios. El propósito de estas pruebas es asegurarse que la función devuelve resultados correctos. Guardar en el archivo `testmydotproduct1.m` y ejecutar:

```
function [] = testmydotproduct2(),
    for testindex = 1 : 10,
        n = round(rand() * 10);
        a = 2 * rand(n, 1) - 1;
        b = 2 * rand(n, 1) - 1;
        res1 = mydotproduct(a, b);
        res2 = a' * b;
        disp(abs(res1 - res2));
    end
end
```

5. Tiempo de ejecución con vectores largos pseudoaleatorios. En algún lenguaje de programación escribir un programa que genere vectores de tamaños grandes: $n = 10^4$, luego $n = 10^5$, luego $n = 10^6$, aplique a estos vectores la función del Ejercicio 1 y mida el tiempo de ejecución. Solución en el lenguaje de Matlab (guardar en el archivo `testmydotproduct3.m`):

```
function [] = testmydotproduct3(),
    nrep = 10;
    for n = [100000, 1000000, 10000000],
        display(n);
        a = rand(n, 1); b = rand(n, 1);
        t1 = cputime();
        for rep = 1 : nrep,
            p = mydotproduct(a, b);
        end
        t2 = cputime();
        display((t2 - t1) / nrep);
    end
end
```

¿Cómo se cambia el tiempo de ejecución cuando n se multiplica por 10?