

CORPORACIÓN UNIVERSITARIA ANTONIO JOSÉ DE SUCRE
FACULTAD DE CIENCIAS DE LA INGENIERIA
INGENIERIA DE SISTEMA

PRESENTADO A:
MSC. JAIDER DE LA ROSA

PRESENTADO POR:
CARLOS COCHERO RAMOS
ANDRES PALACIOS

SINCELEJO, SUCRE

2024

Análisis de eficiencia del algoritmo:

INICIO ALGORITMO

FUNCION sumarParejas (1 OE)

 resultado = 0 (1 OE)

 // Iterar sobre cada elemento del arreglo

 PARA i DESDE 0 HASTA n – 1 INCREMENTAR i+1 HACER (4 OE)

 // Iterar sobre los elementos siguientes al elemento actual

 PARA j DESDE i + 1 HASTA n – 1 INCREMENTAR j+1 HACER (5 OE)

 // Verificar si la suma de los dos elementos es par

 SI (v[i] + v[j]) MODULO 2 ES IGUAL A 0 ENTONCES (5 OE)

 resultado = resultado + 1 (2 OE)

 FIN SI

 FIN PARA

FIN PARA

RETORNAR resultado

FIN FUNCION

FIN ALGORITMO

Entonces tenemos:

$$T_{(n)} = 1 + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} + 1$$

$$T_{(n)} = 1 + 1 \sum_{i=0}^{n-1} (n - i - 1)$$

$$T_{(n)} = 1 + 1 \sum_{i=0}^{n-1} (n - 1) = \frac{n(n-1)}{2}$$

$$T_{(n)} = 2 + 4n + 5 \times \frac{n(n-1)}{2} + 5 \times \frac{n(n-1)}{2} + 2 \times \frac{n(n-1)}{2}$$

$$T_{(n)} = 2 + 4n + 12 \times \frac{n(n-1)}{2}$$

$$T_{(n)} = 2 + 4n + 6n(n-1)$$

$$T_{(n)} = 2 + 4n + 6n^2 - 6n$$

$$T_{(n)} = 6n^2 - 2n + 2$$

Resultados por consola:

Entradas: 10 Tiempo total(ns): 3300

Entradas: 100 Tiempo total(ns): 195700

Entradas: 1000 Tiempo total(ns): 6646100

Entradas: 10000 Tiempo total(ns): 213626300

Entradas: 100000 Tiempo total(ns): 6286061500

Entradas: 1000000 Tiempo total(ns): "Nunca dio resultado"

Resultados por la funcion T(n):

T(10) = 582 OE

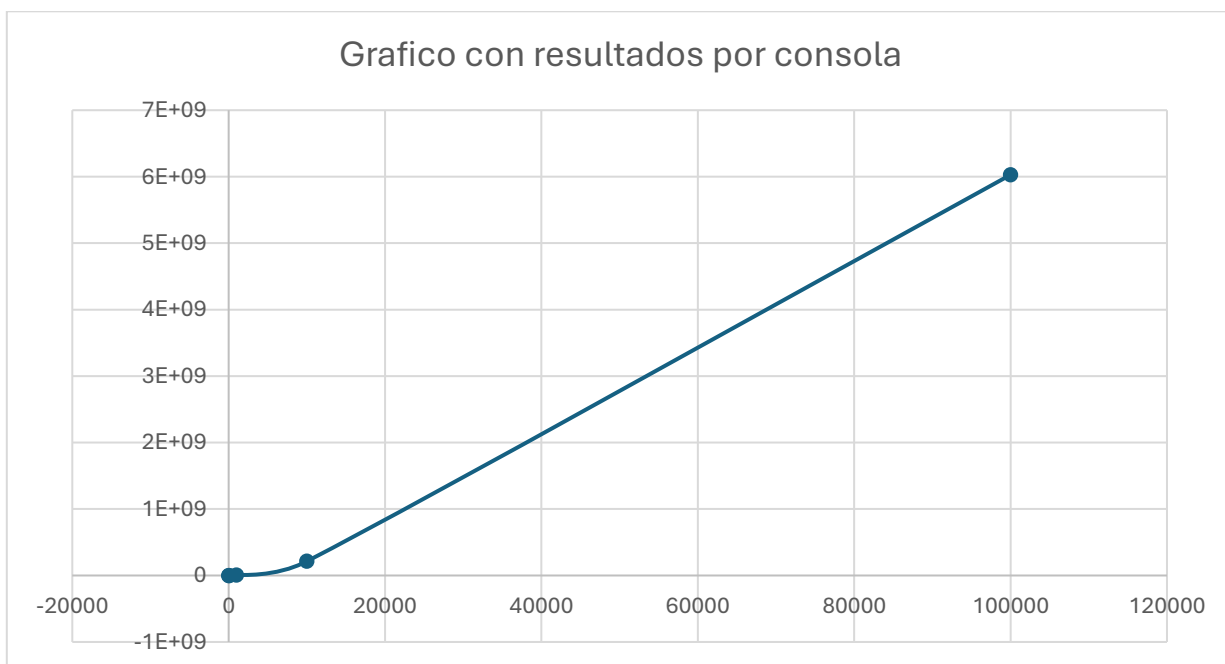
T(100) = 59802 OE

T(1000) = 5998002 OE

T(10000) = 599980002 OE

T(100000) = 59999800002 OE

Graficas de los resultados:



Código JAVA:

```
package actividad2;

public class Actividad2 {

    public static int sumPair(int[] v, int n) {

        long tiempoInicial = System.nanoTime();

        int result = 0;

        for (int i = 0; i < n; i++) {

            for (int j = i + 1; j < n; j++) {

                if ((v[i] + v[j]) % 2 == 0) {

                    result += 1;

                }

            }

        }

        long tiempoFinal = System.nanoTime();

        long tiempoEjecucion = tiempoFinal - tiempoInicial;

        System.out.println("Entradas: " + n + " Tiempo total(ns): " + tiempoEjecucion);

        return result;

    }

    public static void main(String[] args) {

        int n = 1000000;

        int[] v = new int[n];

        for (int i = 0; i < n; i++) {

            v[i] = (int) (Math.random() * 100);

        }

        int result = sumPair(v, n);

        System.out.println(result);

    }

}
```