

PRACTICA 1.1 ALGORITMIA

1. El tiempo medido al ejecutar Vector2 en Java es 0 debido a que los milisegundos antes de ejecutar el método son iguales que los medidos al terminar. Por ello, al realizar la resta el resultado es 0.

Con un tamaño $n = 10000000$ el tiempo que se obtiene ya es válido (porque es mayor que 50 ms) y es de 67 ms.

2. A continuación, se van a comparar los tiempos del algoritmo suma y máximo.

n	Tiempos suma (ms)	Tiempos máximo (ms)
10000	0.067	0.082
20000	0.133	0.161
40000	0.267	0.324
80000	0.56	0.7
160000	1.09	1.39
320000	2.18	2.8
640000	4.33	5.48
1280000	8.9	11.2
2560000	17.8	21
5120000	35.4	44.8
10240000	69	91
20480000	136	169
40960000	273	341
81920000	555	675

Para realizar estas operaciones de manera en que los datos sean válidos, para aquellos valores de n para los que el tiempo de ejecución en milisegundos era menor de 50 se aproximó el valor de una manera distinta: se aumenta el número de repeticiones hasta que el tiempo sea superior a 50 milisegundos y se divide entre el número de repeticiones para pasarlo de nuevo a milisegundos. De esta manera, los valores de la tabla son válidos y fiables.

3. A continuación, se van a comparar los tiempos de los algoritmos que encuentran coincidencias.

n	Tiempos coincidencias1(ms)	Tiempos coincidencias2 (ms)
10000	720	0.078
20000	2592	0.165
40000	10884	0.311
80000	56734	0.67
160000	178505	1.33
320000	FdT	2.69
640000	FdT	5.4
1280000	FdT	11
2560000	FdT	19.5
5120000	FdT	44.9
10240000	FdT	91
20480000	FdT	169
40960000	FdT	341
81920000	FdT	675

En este caso, las mediciones se realizaron como se explicó en el apartado anterior. Para el caso de Coincidencias¹, como está implementado mediante dos bucles anidados (complejidad cuadrática), los tiempos de ejecución son muy grandes y no se realizan todas la mediciones.

La CPU del ordenador que realizó las mediciones es: 11th Gen Intel(R) Core(TM) i7-1165G7
La memoria RAM del ordenador que realizó las mediciones es de: 8,00 GB