TAREA 1

**Nota: las mediciones de este apartado se realizaron con un Mac, que cuenta con un M1**

Modelo procesador: Chip M1

Memoria RAM: 8 GB

Texto, Tabla

Descripción generada automáticamente

El índice medio será de (154+126)/2= 140

Tiempo Benchmarking1 = 212 s

Índice de operaciones reales/enteras = 212 \* 143 = 30313

TAREA 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | CPU | milisegundos | SC MIX (avg) | Operaciones (aprox) |
| 1 | I7-4700U | 285 | 71.3 | 20320,5 |
| 2 | I3-3220 | 267 | 83.3 | 22241,1 |
| 3 | I5-4590 | 219 | 98.1 | 21483,9 |
| 4 | I7-4790 | 207 | 107 | 22149 |
| 5 | Pentium Gold G5400 | 215 | 104 | 22360 |
| 6 | I7-6700K | 303 | 121 | 36.663 |
| 7 | AMD-Ryzen7 4700U | 296 | 105 | 31.080 |

Conclusión: no, porque en función de cada CPU el resultado del algoritmo varía.

ACTIVIDAD 2

-Secuencial

P-alto = 213

P-adaptable = 212

P-bajo = 984

-Paralela

P-alto = 387

P-adaptable = 319

P-bajo = 322

1. El más adecuado es el adaptable después de varias ejecuciones.
2. No, porque la medición ha de ser monótona
3. Si porque así se obtiene una media de tiempo más fiable.

Practica 1.1

**2**

**Nota: las mediciones de este apartado se realizaron con mi ordenador personal, que cuenta con un i7-6700k**

**¿Cuántos años más podremos seguir utilizando esta forma de contar?**

Podremos usar esto hasta el año 584940447

**1. ¿Qué significa que el tiempo medido sea 0?**

Que no es una medida fiable

**2. ¿A partir de qué tamaño de problema (n) empezamos a obtener tiempos fiables?**

130^6

**3 y 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nota: el t en milisegundos | | |  |  |
| *n* | *t suma sin rellenar matriz* | *t máximo sin rellenar matriz* | *T suma con rellenar matriz* | *T máximo con rellenar matriz* |
|  |  |
| 10 | 0 | 0 | 0 | 0 |
| 30 | 0 | 0 | 0 | 0 |
| 90 | 0 | 0 | 0 | 0 |
| 270 | 0 | 0 | 0 | 0 |
| 810 | 0 | 0 | 0 | 0 |
| 2430 | 0 | 0 | 0 | 0 |
| 7290 | 0 | 0 | 0 | 0 |
| 21870 | 0 | 0 | 0 | 0 |
| 65610 | 1 | 1 | 3 | 2 |
| 196830 | 1 | 2 | 5 | 5 |
| 590490 | 1 | 1 | 9 | 65 |
| 1771470 | 0 | 0 | 22 | 19 |
| 5314410 | 2 | 2 | 70 | 81 |
| 15943230 | 6 | 4 | 208 | 222 |
| 47829690 | 17 | 16 | 645 | 630 |
| 143489070 | 55 | 49 | 1904 | 1842 |

1. **¿Qué pasa con el tiempo si el tamaño del problema se multiplica por 5?**

Que el tiempo aumenta de forma lineal.

1. **¿Los tiempos obtenidos son los que se esperaban de la complejidad lineal O(n)?**

Sí que son los esperados.

**6 y 7**

|  |  |  |
| --- | --- | --- |
| n | T diagonal 1 | T diagonal 2 |
| 3 | 0 | 0 |
| 6 | 0 | 0 |
| 12 | 0 | 0 |
| 24 | 0 | 0 |
| 48 | 0 | 0 |
| 96 | 0 | 0 |
| 192 | 1 | 1 |
| 384 | 2 | 3 |
| 768 | 1 | 1 |
| 1536 | 1 | 2 |
| 3072 | 3 | 7 |
| 6144 | 12 | 24 |
| 12288 | 46 | 97 |

A partir del 24576 se queda sin espacio en la memoria.

**¿cumplen los valores obtenidos con lo esperado?**

Sí porque la complejidad esperada es n^2.log(n)

**8**

1. **¿A qué se deben las diferencias de tiempos en la ejecución entre uno y otro programa?**

A que Java es un lenguaje compilado, mientras que Python es un lenguaje interpretado.

1. **Independientemente de los tiempos concretos ¿existe alguna analogía en el comportamiento de las dos implementaciones?**

Crecen de forma similar, pero con tiempos claramente distintos.