

Metodos de ordenamiento en Python comparativa del metodo tradicional y con hilos

Cristhian Urrego Salazar - cristhianu92@gmail.com, Andres Felipe Farfan - anfefarfan@hotmail.com, and heriberto david yepes - HDYCOTECNOVA@gmail.com

I. RESUMEN

Buscando averiguar la eficacia en mtodos de ordenamiento experimentamos los resultados en los algoritmos al implementar hilos al cdigo usando python como lenguaje para subdividir los mtodos en tareas simultaneas con la intencin de averiguar si lograba mejorar su rendimiento.

II. PALABRAS CLAVES

Python, Ordenamiento, InsertionSort, MergeSort, HeapSort, QuickSort, CoutingSort, RadixSort, Github, Hilos

III. RESUMEN EN INGLES

Find the efficiency in ordering methods to experiment the results in the algorithms to implement threads in the code, using python as a language to subdivide the methods in simultaneous tasks with the intention of finding out if it is to improve their performance

IV. PALABRAS CLAVES EN INGLES

Python, Ordering, InsertionSort, MergeSort, HeapSort, QuickSort, CoutingSort, RadixSort, Github, Threads

V. INTRODUCCION

Buscando averiguar la eficacia en mtodos de ordenamiento experimentamos los resultados en los algoritmos al implementar hilos al cdigo usando python como lenguaje para subdividir los mtodos en tareas simultaneas con la intencin de averiguar si lograba mejorar su rendimiento.

VI. METODOLOGIA

Al ser un proyecto pequeno se uso programacin extrema como metodologia

VII. ALGORITMOS USADOS

InsertionSort, MergeSort, HeapSort, QuickSort, CoutingSort, RadixSort

VIII. LENGUAJE DE PROGRAMACIN USADO

Python es un lenguaje de programacin interpretado cuya filosofia hace hincapi en una sintaxis que favorezca un cdigo legible.

Se trata de un lenguaje de programacin multi-paradigma, ya que soporta orientacin a objetos, programacin imperativa y, en menor medida, programacin funcional. Es un lenguaje interpretado, usa tipado dinmico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de cdigo abierto, denominada Python Software Foundation License,² que es compatible con la Licencia pblica general de GNU a partir de la versin 2.1.1, e incompatible en ciertas versiones anteriores. (Extraido de wikipedia)

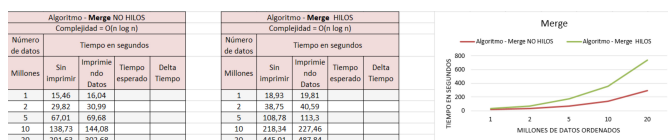
IX. CODIGO DE ALGORITMOS IMPLEMENTADO

Repositorio Github
https://github.com/afurourrego/ordenamientos_python/blob/master/menu.py

X. RESULTADOS ALCANZADOS

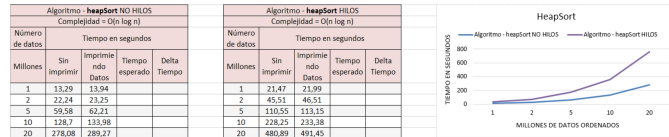
Algoritmo - InsertionSort NO HILOS					Algoritmo - InsertionSort HILOS				
Complejidad = $O(n^2)$					Complejidad = $O(n^2)$				
Número de datos	Tiempo en segundos				Número de datos	Tiempo en segundos			
Millones	Sin imprimir	Imprimiendo Datos	Tiempo esperado	Delta Tiempo	Millones	Sin imprimir	Imprimiendo Datos	Tiempo esperado	Delta Tiempo
1	se corrio el algoritmo durante 4 horas sin resultados terminados.				1	no se realizo el proceso con hilos ya que se evidencia que toma mas tiempo en este tipo de procesos de ordenamiento.			
2					2				
5					5				
10					10				
20					20				

https://raw.githubusercontent.com/afurourrego/ordenamientos_python/master/a1_insertionsort.jpg

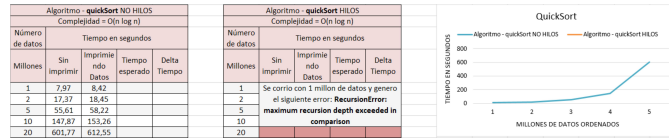


<https://raw.githubusercontent.com/afurourrego/>

ordenamientos_phyton/master/a2_merge.jpg



https://raw.githubusercontent.com/afurourrego/ordenamientos_phyton/master/a3_heapsort.jpg



https://raw.githubusercontent.com/afurourrego/ordenamientos_phyton/master/a4_quicksort.jpg

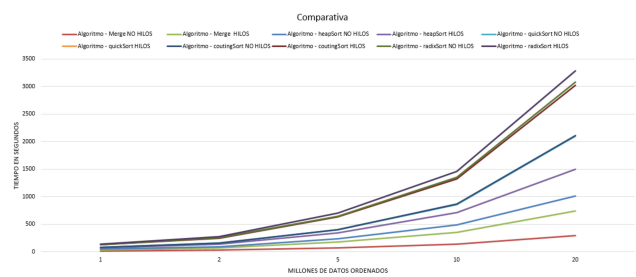


https://raw.githubusercontent.com/afurourrego/ordenamientos_phyton/master/a5_countingsort.jpg



https://raw.githubusercontent.com/afurourrego/ordenamientos_phyton/master/a6_radixsort.jpg

XI. EVALUACION DE LOS RESULTADOS DE LOS ALGORITMOS IMPLEMENTADOS



https://raw.githubusercontent.com/afurourrego/ordenamientos_phyton/master/a6_radixsort.jpg

ordenamientos_phyton/master/a0_comparativa.jpg

XII. CONCLUSIONES

- El mtodo de insertionSort es obsoleto para ordenamientos en millones de datos.
- El Mejor mtodo ms eficaz en este tipo de ordenamientos es CountingSort.
- La aplicacin de Threads para este caso no hace el trabajo ms eficiente.

XIII. RECOMENDACIONES

- Para ordenamiento de nmeros se recomienda usar CountingSort al ser el mas eficiente.

XIV. COMPUTADOR USADO

- Macbook Pro 2014
- Procesador 2.6 GHz Intel Core i5
- Memoria 8 GB 1600 MHz DDR3
- Graficos Intel Iris 1536 MB

XV. REFERENCIAS BIBLIOGRAFICAS

Python, 2019, <https://es.wikipedia.org/wiki/Python>