Show/Hide Code Button

Click on the button above to show/hide code.

Algoritmos de ordenamiento

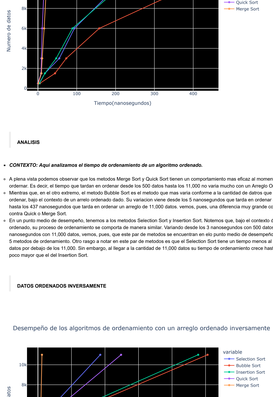
Mauricio Vazquez Moran
000191686
Estructuras de Datos Avanzadas
2022-09-19

OBJETIVO

El objetivo de esta tarea es determinar empíricamente el desempeño de los algoritmos de ordenamiento vistos en clase.

ANALISIS TIEMPO DE ORDENAMIETO

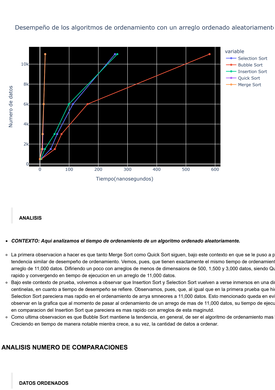
DATOS ORDENADOS



ANALISIS

- ***CONTEXTO: Aquí analizamos el tiempo de ordenamiento de un algoritmo ordenado.***
- A plena vista podemos observar que los metodos Merge Sort y Quick Sort tienen un comportamiento mas eficaz al momento de ordenar. Es decir, el tiempo que tardan en ordenar desde los 500 datos hasta los 11,000 no varia mucho con un Arreglo Ordenado.
- Mientras que, en el otro extremo, el metodo Bubble Sort es el metodo que mas varia conforme a la cantidad de datos que se le piden ordenar, bajo el contexto de un arreglo ordenado dado. Su variacion viene desde los 5 nanosegundos que tarda en ordenar 500 datos hasta los 437 nanosegundos que tarda en ordenar un arreglo de 11,000 datos. vemos, pues, una diferencia muy grande comparando contra Quick o Merge Sort.
- En un punto medio de desempeño, tenemos a los metodos Selection Sort y Insertion Sort. Notemos que, bajo el contexto de un arreglo ordenado, su proceso de ordenamiento se comporta de manera similar. Variando desde los 3 nanosegundos con 500 datos 237 nanosegundos con 11,000 datos, vemos, pues, que este par de metodos se encuentran en ete punto medio de desempeño de nuestros 5 metodos de ordenamiento. Otro rasgo a notar en este par de metodos es que el Selection Sort tiene un tiempo menos al ordenar datos por debajo de los 11,000. Sin embargo, al llegar a la cantidad de 11,000 datos su tiempo de ordenamiento crece hasta ser un poco mayor que el del Insertion Sort.

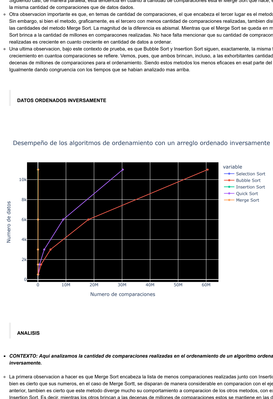
DATOS ORDENADOS INVERSAMENTE



ANALISIS

- ***CONTEXTO: Aquí analizamos el tiempo de ordenamiento de un algoritmo ordenado inversamente.***
- La primera observacion que podemos hacer, en este caso, es que Quick Sort y Merge Sort dejan de ser ramillas en cuanto a tiempo de ordenamiento. Vemos, pues, que la caracteristica que habiamos encontrado anteriormente ya no es. En cambio, Merge Sorte sigue una tendencia de bajo tiempo de ordenamiento de datos. Mientras que, Quick Sort, se divierge de esta tendencia que habia tenido anteriormente y, incluso, se vuelve mas lento en el ordenamiento que Selection Sort.
- Por otro lado, algo que cabe mencionar es que Selection Sort aquí parecieria ser mas eficaz. Es decir, bajo un contexto de ordenar un arreglo que esta ordenado inversamente. Vemos, pues, que sus tiempos mejoran en relacion con el ejercicio pasado. Esto es desde los 0 nanosegundos para ordenar 500 datos hasta los 186 nanosegundos para ordenar 11,000 datos. No podemos negar una mejora en su rendimiento.
- Un rasgo distintivo y permanente en el ordenamiento del Bubble Sort es que es el mas lento de los 5 metodos de ordenamiento. Notas que bajo otro diferente contexto difiere, con gran magnitud de tiempo de los otros metodos de ordenamiento. Agregando que, bajo este contexto de test de ordenamiento, su centinela, en cuanto a tiempo de desempeño se refire, es el metodo Insertion Sort. Siendo, pues, el Insertion Sort un poco mas rapido que Bubble Sorte pero siguiendo con la misma tendencia de crecimiento de tiempo.

DATOS ORDENADOS ALEATORIAMENTE



ANALISIS

- ***CONTEXTO: Aquí analizamos el tiempo de ordenamiento de un algoritmo ordenado aleatoriamente.***
- La primera observacion a hacer es que tanto Merge Sort como Quick Sort siguen, bajo este contexto en que se le puso a prueba, una tendencia similar de desempeño de ordenamiento. Vemos, pues, que tienen exactamente el mismo tiempo de ordenamiento con un arreglo de 11,000 datos. Difriendo un poco con arreglos de menos de dimensaions de 500, 1,500 y 3,000 datos, siendo Quick Sort mas rapido y convergendo en tiempo de ejecucion en un arreglo de 11,000 datos.
- Bajo este contexto de prueba, volvemos a observar que Insertion Sort y Selection Sort vuelven a verse inmersos en una dinamica de centinelas, en cuanto a tiempo de desempeño se refiere. Observamos, pues, que, al igual que en la primera prueba que hicimos, Selection Sort pareceria mas rapido en el ordenamiento de arrya sinmeores a 11,000 datos. Esto mencionado queda en evidencia, al observar en la grafica que al momento de pasar al ordenamiento de un arrego de mas de 11,000 datos, su tiempo de ejecucion reporta en comparacion del Insertion Sort que pareciera es mas rapido con arreglos de esta magnitud.
- Como ultima observacion es que Bubble Sort mantiene la tendencia, en general, de ser el algoritmo de ordenamiento mas lento. Creciendo en tiempo de manera notable mientra crece, a su vez, la cantidad de datos a ordenar.

ANALISIS NUMERO DE COMPARACIONES

DATOS ORDENADOS

ANALISIS

- ***CONTEXTO: Aquí analizamos la cantidad de comparaciones realizadas en el ordenamiento de un algoritmo ordenado.***
- El primer rasgo importante a notar es que Selection Sort, bajo un contexto de un arreglo ya ordenado, hace, el mismo numero de comparaciones que de datos menos uno. Siendo, es decir, metodo de ordenamiento que compara menos en su camino a ordenar. Siguiendo casi, de manera paralela, esta tendencia en cuanto a cantidad de comparaciones esta el Merge Sort que hace, exactamente, la misma cantidad de comparaciones que de datos dados.
- Otra observacion importante es que, en temas de cantidad de comparaciones, el que encabeza el tercer lugar es el metodo Quick Sort. Sin embargo, si bien el metodo, graficamente, es el tercero con menos cantidad de comparaciones realizadas, tambien dista mucho de las cantidades del metodo Merge Sort. La magnitud de la diferencia es abismal. Mientras que el Merge Sort se queda en miles, Quick Sort brinca a la cantidad de millones en comparaciones realizadas. No hace falta mencionar que su cantidad de comparaciones realizadas es creciente en cuanto creciente en cantidad de datos a ordenar.
- Una ultima observacion, bajo este contexto de prueba, es que Bubble Sort y Insertion Sort siguen, exactamente, la misma tendecia de crecimiento en cuantas comparaciones se refiere. Vemos, pues, que ambos brincan, incluso, a las exhorbitantes cantidades de decenas de millones de comparaciones para el ordenamiento. Siendo estos metodos los menos eficaces en esat parte del analisis. Igualmente dando congruencia con los tiempos que se habian analizado mas arriba.

DATOS ORDENADOS INVERSAMENTE

ANALISIS

- ***CONTEXTO: Aquí analizamos la cantidad de comparaciones realizadas en el ordenamiento de un algoritmo ordenado inversamente.***
- La primera observacion a hacer es que Merge Sort encabeza la lista de menos comparaciones realizadas junto con Insertion Sort. Si bien es cierto que sus numeros, en el caso de Merge Sortt, se disparan de manera considerable en comparacion con el ejercicio anterior, tambien es cierto que este metodo diverge mucho su comportamiento a comparacion de los otros metodos, con excepcion del Insertion Sort. Es decir, mientras los otros brincan a las decenas de millones de comparaciones estos se mantiene en las decenas de miles. En especifico, el Insertion Sort se mantiene a la par de datos y comparaciones, es decir, casi las mismas cantidades.
- La segunda observacion, quizá la mas notoria, es que, bajo este criterio de pruebas, tanto Selection Sort como Quick Sort brincan ya, pues, a las decenas de millones de comparaciones. Situacion que no se habia visto en el ejercicio anterior del analisis. Siendo, pues, Selection Sort el que cnoce en mayor cantidad de comparaciones llegando hasta la cantidad de casi 60.5 millones de comparaciones para el ordenamiento de 11,000 datos. Mientras que el Quick Sort, en su extremo mas alto llega la cantida de 30.25 millones de comparaciones para el ordenamiento de dicha cantidad.
- Una ultima observacion a esta parte del analisis es que el Bubble Sort mantiene, hasta el momento y bajos ya los dos contexto de ejercicios analizados, una tendencia de cantidad de comparaciones en decenas de millones. Siendo, pues, el metodo que presenta mas cantidad de comparaciones hasta el momento.

DATOS ORDENADOS ALEATORIAMENTE

ANALISIS

- ***CONTEXTO: Aquí analizamos la cantidad de comparaciones realizadas en el ordenamiento de un algoritmo ordenado aleatoriamente.***
- La primera observacion característica de esta parte del analisis es que los metodos Merge Sort y Quick Sort siguen una dinamica de centinelas en cuanto a cantidad de comparaciones se refiere. En otras palbras, su cantidad de comparaciones no dista mucho y siguen un dinamica de crecimiento similar, bajo sus respectivas cantidades de datos. Siendo, pues, el Quick Sort el metodo que presenta menor cantidad de comparaciones.
- Otra caracteristica notoria, bajo este contexto de analisis, es que el metodo Insertion Sort mantiene un dinamica de crecimiento en cantidad de comparaciones afin, en comparacion con Quick Sort y Merge Sort. Sin embargo, no presenta la cantidad de comparaciones que si presentan los centinelas en Selection Sort y Bubble Sort. Notamos, pues, que es como el punto medio de cantidad de comparaciones. Bajo este contexto.
- Como se ha prensentando hasta ahorita Bubble Sort prenseta una tendencia alta en cuant a cantidad de comparaciones en el proceso de ordenamiento. A su vez, el metodo Selection Sort, se le une en, exactamente, la misma cantidad y tendencia de comparaciones. Siendo, pues, este par de metodos los que tienen una mayor cantidad de comparaciones bajo este contexto de analisis.