

Comparativa de Algoritmos de Ordenamiento

Luis Borit Guitton - Abel Borit Guitton - Jesus Alpaca Rendon
Maestria en Ciencias de la Computacion, Universidad Nacional de San Agustin

2023
Julio

1 Introduccion

El presente informe muestra la comparativa de tiempos de ejecucion realizada entre 5 algoritmos de ordenamiento como son: Binary Insertion Sort, Bubble sort, Quick sort, Selection sort y Merge sort. Implementados en 3 diferentes lenguajes de programacion: Python, Golang y C++. Los algoritmos fueron sometidos a pruebas para obtener un tiempo promedio de ejecucion y ver los resultados en una grafica comparativa

2 Algoritmos

Los algoritmos seleccionados para el presente trabajo de investigacion son los siguientes:

- * Binary Tree
- * Bubble sort
- * Quick sort
- * Selection sort
- * Merge Sort

1. Binary Insertion Sort
2. Bubble Sort

Este algoritmo realiza el ordenamiento o reordenamiento de una lista a de n valores, en este caso de n términos numerados del 0 al n-1; consta de dos bucles anidados, uno con el índice i, que da un tamaño menor al recorrido de la burbuja en sentido inverso de 2 a n, y un segundo bucle con el índice j, con un recorrido desde 0 hasta n-i, para cada iteración del primer bucle, que indica el lugar de la burbuja. La burbuja son dos términos de la lista seguidos, j y j+1, que se comparan: si el primero es mayor que el segundo sus valores se intercambian.

Data: $a_1, a_2, a_3, a_4 \dots a_{(n-1)}$
Result: ordered list
 initialization;
for i **to** $n-1$ **do**
 for j **to** $n-i-1$ **do**
 if $a_j > a_{j+1}$ **then**
 go to next section;
 $aux \leftarrow a_j$;
 $a_j \leftarrow a_{j+1}$;
 $a_{j+1} \leftarrow aux$;
 end
 end
end

Algorithm 1: Bubble Algorithm

3. Quick Sort
4. Selection Sort
5. Merge Sort

3 Implementacion

4 Resultados

5 Conclusiones