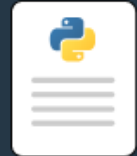


Go vs. Python and C++ – main differences



codilime
CREATING VALUE



PRACTICA 12

Fundamentos de lenguaje de programacion

ESTUDIANTE:

Roger Euclides Infa Sanchez

a

b

DOCENTE:

Vicente Enrique Machaca Arceda



1 Estructura

Parte	Responsable
Codigo Go	Frank Duarte
Codigo c++	Guillermo Caceres
Codigo Python	Roger Infa
Codigo Grafico	Roger Infa
Informe	Roger Infa

2 Introduccion

En esta práctica se buscará implementar satisfactoriamente dos algoritmos de ordenamiento, uno con tiempo computacional cuadrático y el otro con tiempo computacional logarítmico, esto con el fin de comprobar que en algunos casos el uso de uno u otro método puede beneficiarnos hablando de costos computacionales ya sea para una cantidad pequeña de datos o también para cantidades grandes. Además de ello, se compararán los tiempos de ejecución entre tres lenguajes de programación, estos son python, c++ y go.

Para la implementación de esta práctica se definirá el hardware que usé para hacer todas las pruebas empezando por el Sistema Operativo, en este caso es Windows 11. De procesador tengo un Intel 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz 2.69 GHz acompañado de 16.0 GB (15.8 GB usable) de RAM que trabajan con una frecuencia de 3.0GHz, con un disco sólido SSD de 512GB .

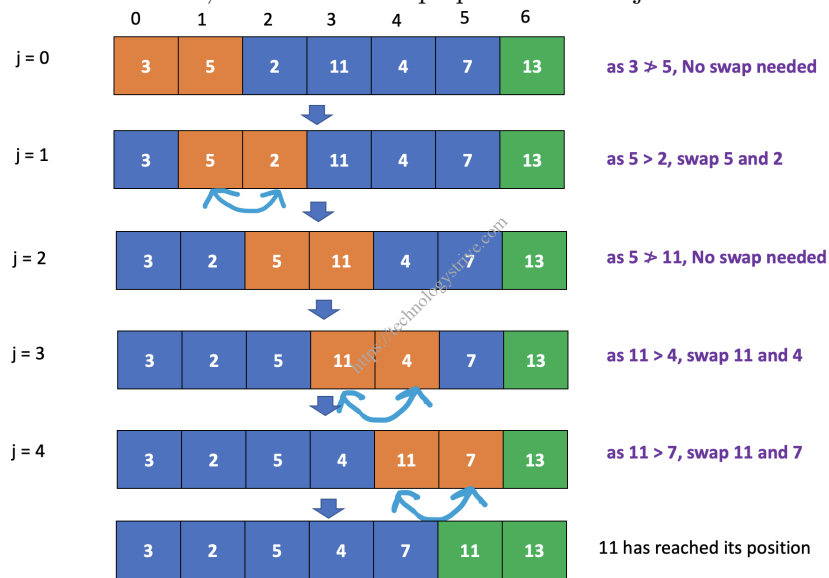
Estas prácticas fueron realizadas en los lenguajes de programación Python, c++ y go. Además de ello, se escogió Python para la generación de gráficos porque es fácil de entender e implementar gráficos y es de tipo dinámico que en lo personal es fabuloso trabajar con un lenguaje así.

Para implementar los gráficos se tuvo que instalar las librerías numpy y matplotlib con el comando **pip install NOMBRE LIBRERIA**

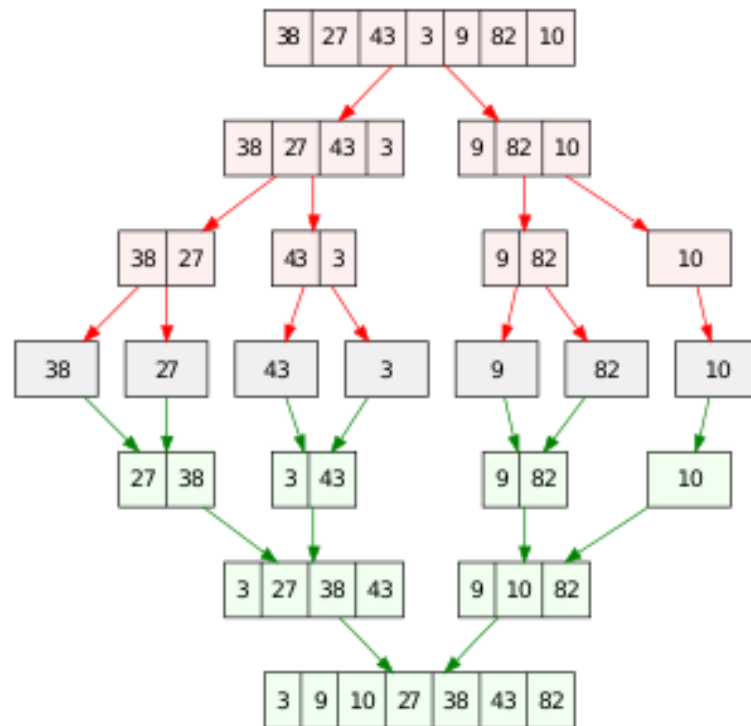
3 Algoritmos (los 02 algoritmos que va a implementar)

Para hacer las comparaciones, hemos implementado un algoritmo de costo cuadrático y otro de costo logarítmico, estos son el bubble sort y merge sort respectivamente.

El bubble sort o también llamado ordenamiento de burbuja funciona marcando cada elemento de la lista que se ordenará con el siguiente elemento y en cada "burbuja" intercambiara los elementos si es mayor que. Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran pequeñas "burbujas".



El algoritmo Merge Sort es un algoritmo de clasificación que se basa en el paradigma divide y conquistaras. En este algoritmo, la matriz se divide inicialmente en dos mitades iguales y luego se combinan de manera ordenada utilizando comparaciones y recursividad.



4 Implementacion(codigo fuente)

5 Resultados

5.1 Tabla comparativa con el promedio de tiempo de procesamiento y desviacion estandar

5.2 Graficos

6 Conclusiones

SOLUCION:

```
// codigo
```