

CARRERA: Computación

ASIGNATURA: Estructura de Datos

NRO. PRÁCTICA: 4.2 **TÍTULO PRÁCTICA:** Algoritmos de Ordenamiento

OBJETIVO (Colocar el o los objetivos que se alcanzarán al desarrollar la práctica):

- Relaciona las técnicas de ordenamiento.
- Desarrolla módulos que emplean técnicas de ordenamiento.

INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):

1. Generar una aplicación que por cada método de ordenamiento (burbuja con ajuste, selección e inserción,) y por cada grupo de valores (n cantidad de números aleatorios) presente el tiempo de ejecución.

Para este ejercicio no imprimir la ejecución del método de ordenamiento, sólo el tiempo que se demoró en su resolución.

Tomar el tiempo antes de ejecutar el algoritmo y despues para obtener el tiempo de ejecución

Los grupos de valores que se usarán para las pruebas son:

- 5.000
- 10.000
- 30.000
- 50.000
- 100.000

NOTA : se debe usar el mismo para arreglar con otro metodo, considerar usar clone o copiar la instancia, de no estara ya ordenaro, en el caso de mandar a arreglar con el segundo metodo un arreglo que ya esta ordenado se considerara al metodo como incorrecto.

2. Generar un arreglo aleatorio que sea el mismo para todas las llamadas al metodo de ordenamiento, es decir el arreglo de tamaño 10 debera ser los primeros 10 numeros del tamaño del arreglo 20 y asi en adelante.

EJEMPLOS:

Arreglo de tamaño 10:

[23, 405, 12, 28001, 34, 5006, 78, 90, 11, 4443]

Arreglo de tamaño 100:

Los primeros 10 números serán los mismos del arreglo anterior y luego se añadirán 10 números adicionales.

[23, 405, 12, 28001, 34, 5006, 78, 90, 11, 4443, 67, 89, 24, 48, 71, 13, 59, 31, 62, 84,]

Con los datos generados se solicita realizar un informe (adjunto se encuentra el formato) donde se presente las capturas de pantalla donde se muestre el tiempo que se demoró en ordenar los valores por cada cantidad de números y por método de ordenamiento.

Adicional, incluir una gráfica por método donde se presente en el eje X la cantidad de valores generados y en el eje Y el tiempo utilizado por cantidad de valores. Para esta gráfica usar un gráfico de líneas donde se evidencie todos los metodos en un solo gráfico.

Informe debera contener:

- ENLACE AL REPOSITIO GIT DEL POREYCTO

1. Criterios de valoración: **NO SE CALIFICARÁ** si el desarrollo de los métodos de ordenamiento no está funcionando, puesto que como pueden obtener el tiempo de ejecución de un método de ordenamiento que no funciona.
2. **NO SE CALIFICARA** si el arreglo que llega al metodo ya fue ordenado previamente.

Criterio	Puntaje
Aplicación	
Ejecución de app.py y muestre en consola todos los resultados	4
Informe README con las tablas, capturas de grafica y conclusiones	
• Gráfica con todos los metodos	2
• Tabla de resultados todos los metodos	2
• Analisis de resultados (Conclusiones) POR ESTUDIANTE	2

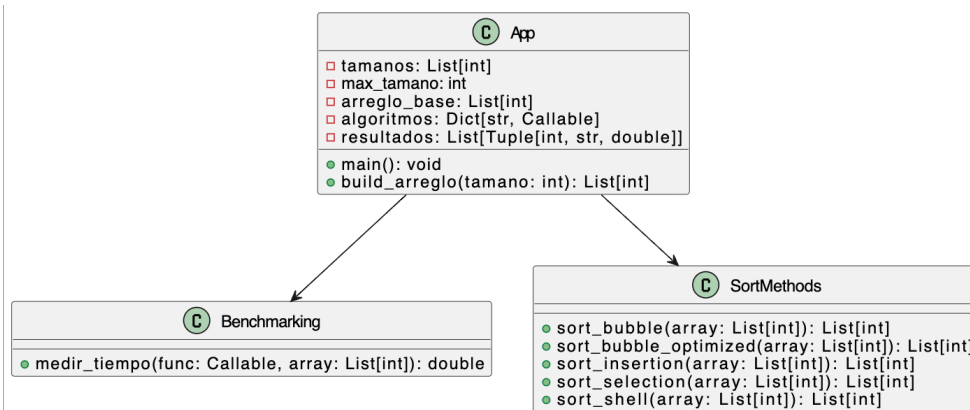
4. Consideraciones:

- La aplicación se realizará en parejas y se subirá al AVAC
- En caso de copia, los grupos tendrán calificación de cero.

ACTIVIDADES POR DESARROLLAR

(Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)

1. Diagrama de clase tentativo



2. Generar una aplicación que determine la complejidad de los algoritmos de ordenamiento Burbuja , Burbuja con Ajuste, Selección, Inserción y Shell.

Para lo cual, se solicita por método de ordenamiento generar números aleatorios con la cantidad de valores que se indica en las instrucciones.

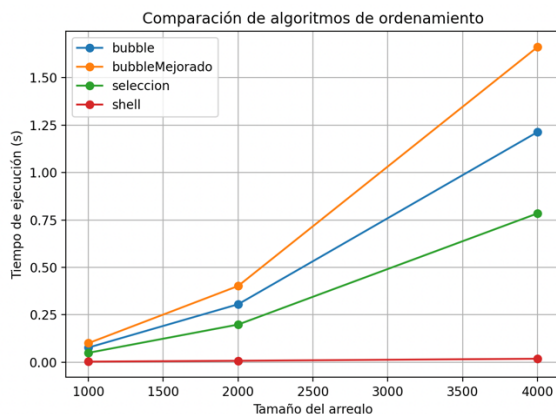
Salida de consola de la ejecución esperada

Tamano: 1000, Algoritmo: bubble, Tiempo: 0.021 segundos

Tamano: 1000, Algoritmo: bubbleMejorado, Tiempo: 0.0287 segundos
 Tamano: 1000, Algoritmo: seleccion, Tiempo: 0.0139 segundos
 Tamano: 1000, Algoritmo: insercion, Tiempo: 0.0139 segundos
 Tamano: 1000, Algoritmo: shell, Tiempo: 0.0011 segundos
 Tamano: 2000, Algoritmo: bubble, Tiempo: 0.0827 segundos
 Tamano: 2000, Algoritmo: bubbleMejorado, Tiempo: 0.1209 segundos
 Tamano: 2000, Algoritmo: seleccion, Tiempo: 0.0555 segundos
 Tamano: 2000, Algoritmo: insercion, Tiempo: 0.0139 segundos
 Tamano: 2000, Algoritmo: shell, Tiempo: 0.0027 segundos
 Tamano: 4000, Algoritmo: bubble, Tiempo: 0.3415 segundos
 Tamano: 4000, Algoritmo: bubbleMejorado, Tiempo: 0.51 segundos
 Tamano: 4000, Algoritmo: seleccion, Tiempo: 0.224 segundos
 Tamano: 4000, Algoritmo: insercion, Tiempo: 0.0139 segundos
 Tamano: 4000, Algoritmo: shell, Tiempo: 0.0063 segundos

ESTE ES UN EJEMPLO LOS VALORES DEBERIAN SER DISTINTOS

En el informe debera estar el parrafo de Conclusiones al analisis de las gráficas presentadas. Utilizar terminologia de Notación y Complejidad



En el informe debera incluir enlace al repositorio en GITHUB o GITLAB del proyecto, se revisara el ultimo commit hasta antes de la fecha de entrega.

RESULTADO(S) OBTENIDO(S):

Escribir los resultados obtenidos con la realización de la práctica.

- Conocimiento de los Algoritmos de Ordenamiento.
- Determinación de la diferencia de cada algoritmo de ordenamiento.

CONCLUSIONES:

- Cada estudiante conocerá el procedimiento para los Algoritmos de Ordenamiento.
- Cada estudiante identificará el algoritmo más eficiente en base al número de iteraciones.

RECOMENDACIONES:

- Revisar el material de la sección "Contenido Teórico"
- Se debe desarrollar en el lenguaje de programación JAVA o Python

Docente / Técnico Docente: Ing. Pablo Torres

Firma: _____

