Laboratorio N-3.2

Antes de comenzar ...

1. Instalar dependecias y creacion del `package.lock.json`

   ```
   npm i
   ```

2. El codigo fue separado en carpetas `functions` y `clases` su funcion tiene una relacion directa con el nombre respectivo.
3. Compilar el programa

   ```
   node index.js > data/data.csv
   ```

   La salida de consola sera almacenada en un archivo `.csv`

Intercalacion vs merge

- ***Algoritmo de intercalacion***

  El algoritmo de intercalacion espera recibir un arreglo con sub arreglos ordenados, como se vio en clases

- ***Algoritmo de merge***

  El merge(arr, l, m, r) es un proceso clave que asume que arr[l.. m] y arr[m+1..r] se ordenan y fusionan las dos sub-matrices ordenadas en una sola.

Pros y contras

Intercalacion

Pros

- Es un algoritmo que arregla dado dos subarreglos ordenados
- Tiene menos lineas de codigo

Contra

- Dado que recibe un arreglo entero podemos suponer que pudiera tener una complejidad lineal
- Tenemos a usar ciclos iteradores lo que significa un coste asintotico de O(n)
- Tiene mayor costo de tiempo
- Algoritmo de fuerza bruta

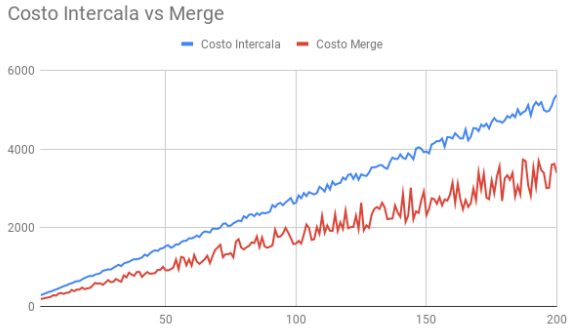
Merge

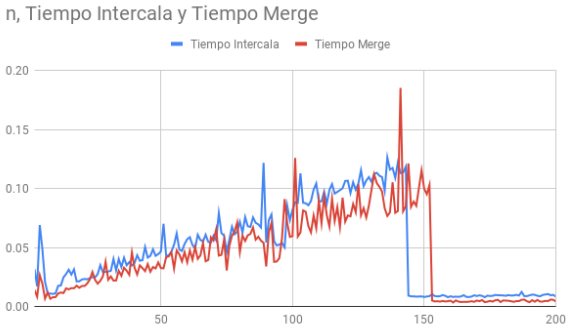
Pros

- Combina dos subarreglos pero de una funcion exterior recursiva `MergeSort`
- Aplica el metodo de programacion ***"divide y venceras"***, si bien ejecuta muchas lineas de codigo tiene un coste menor dado a que con el concepto antes mencionado se busca la optimizacion del codigo

Contra

- Tiene muchas lineas de codigo
- Merge no ordena un algoritmo de forma independiente nesecita arreglos recursivos auxiliares
- Creacion de nuevos intancias y memorias para creacion de arreglos





Conclusion

Podriamos decir que el ***Intercalation*** es un algoritmo que aplica el metodo de "fuerza bruta" pero con el paso del tiempo y aplicacion de metodologias modernas para la optimacion el ***Merge*** es una mejora de ***Intercalation.***