



Tecnológico de Monterrey

**INSTITUTO TECNOLÓGICO Y
DE ESTUDIOS SUPERIORES DE MONTERREY**

Programación de estructuras
de datos y algoritmos fundamentales

Act 1.2 - Algoritmos de Búsqueda y Ordenamiento

Profesor:

Mauricio Paletta Nannarone

Alumno: Dael Chávez Ferreyra - A01771963

12, septiembre 2023

- El orden (O) del programa se puede analizar de la siguiente manera:

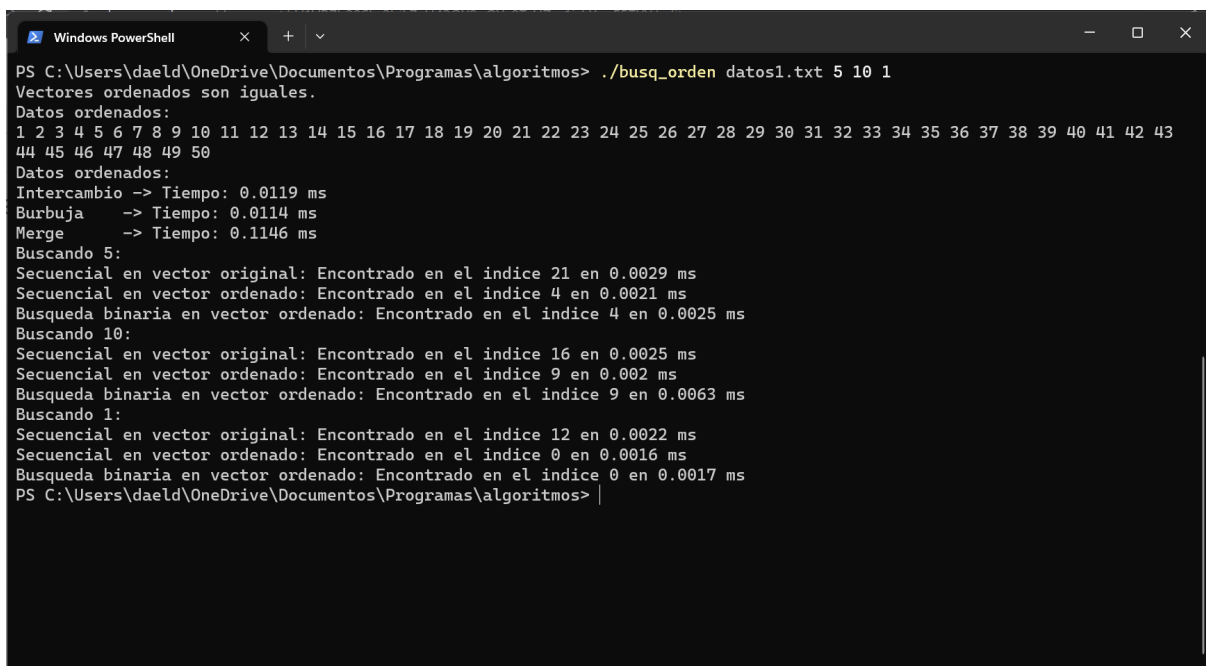
- Cargar los enteros desde el archivo: La carga de enteros desde un archivo se realiza en $O(n)$ en el peor de los casos, donde "n" es el número de enteros en el archivo.
- Tres algoritmos de ordenamiento: El programa utiliza tres algoritmos de ordenamiento: Intercambio, Burbuja y Mergesort. El ordenamiento por Intercambio y Burbuja tiene una complejidad de $O(n^2)$ en el peor de los casos, mientras que Mergesort tiene una complejidad de $O(n \log n)$ en el peor de los casos.
- Búsqueda secuencial y búsqueda binaria: La búsqueda secuencial tiene una complejidad de $O(n)$ en el peor de los casos, mientras que la búsqueda binaria tiene una complejidad de $O(\log n)$ en el peor caso, pero requiere que el vector esté ordenado previamente.

Entonces, el mayor orden (O) del programa es n^2 .

Ahora, presentaré 6 ejemplos de ejecuciones del programa con 3 distintos archivos:

- Pruebas con el archivo "datos1.txt", que tiene 50 enteros desordenados:

1. **./busq_orden datos1.txt 5 10 15**: El programa ordenará los enteros y luego realizará búsquedas de los números 5, 10 y 15.



```
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos1.txt 5 10 1
Vectores ordenados son iguales.
Datos ordenados:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
Datos ordenados:
Intercambio -> Tiempo: 0.0119 ms
Burbuja -> Tiempo: 0.0114 ms
Merge -> Tiempo: 0.1146 ms
Buscando 5:
Secuencial en vector original: Encontrado en el indice 21 en 0.0029 ms
Secuencial en vector ordenado: Encontrado en el indice 4 en 0.0021 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 4 en 0.0025 ms
Buscando 10:
Secuencial en vector original: Encontrado en el indice 16 en 0.0025 ms
Secuencial en vector ordenado: Encontrado en el indice 9 en 0.002 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 9 en 0.0063 ms
Buscando 1:
Secuencial en vector original: Encontrado en el indice 12 en 0.0022 ms
Secuencial en vector ordenado: Encontrado en el indice 0 en 0.0016 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 0 en 0.0017 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> |
```

2. **./busq_orden datos1.txt 5**: El programa ordenará los enteros y luego realizará la búsqueda del número 5.

```
Windows PowerShell
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos1.txt 5
Vectores ordenados son iguales.
Datos ordenados:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50
Datos ordenados:
Intercambio -> Tiempo: 0.0108 ms
Burbuja -> Tiempo: 0.0103 ms
Merge -> Tiempo: 0.1112 ms
Buscando 5:
Secuencial en vector original: Encontrado en el indice 21 en 0.0046 ms
Secuencial en vector ordenado: Encontrado en el indice 4 en 0.0018 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 4 en 0.0017 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> |
```

- Pruebas con el archivo "datos2.txt", que tiene 100 enteros desordenados:

1. **./busq_orden datos2.txt 7 20 30**: El programa ordenará los enteros y luego realizará búsquedas de los números 7, 20 y 30.

```
Windows PowerShell
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos2.txt 7 20 30
Vectores ordenados son iguales.
Datos ordenados:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
Datos ordenados:
Intercambio -> Tiempo: 0.041 ms
Burbuja -> Tiempo: 0.0407 ms
Merge -> Tiempo: 0.2314 ms
Buscando 7:
Secuencial en vector original: Encontrado en el indice 51 en 0.0023 ms
Secuencial en vector ordenado: Encontrado en el indice 6 en 0.0018 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 6 en 0.0019 ms
Buscando 20:
Secuencial en vector original: Encontrado en el indice 63 en 0.002 ms
Secuencial en vector ordenado: Encontrado en el indice 19 en 0.0017 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 19 en 0.0018 ms
Buscando 30:
Secuencial en vector original: Encontrado en el indice 34 en 0.002 ms
Secuencial en vector ordenado: Encontrado en el indice 29 en 0.0018 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 29 en 0.0018 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> |
```

2. **./busq_orden datos2.txt 7:** El programa ordenará los enteros y luego realizará la búsqueda del número 7.

```
Windows PowerShell
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos2.txt 7
Vectores ordenados son iguales.
Datos ordenados:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
Datos ordenados:
Intercambio -> Tiempo: 0.0411 ms
Burbuja -> Tiempo: 0.0406 ms
Merge -> Tiempo: 0.2108 ms
Buscando 7:
Secuencial en vector original: Encontrado en el indice 51 en 0.0024 ms
Secuencial en vector ordenado: Encontrado en el indice 6 en 0.0052 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 6 en 0.0027 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos>
```

- Pruebas con el archivo "datos3.txt", que tiene 150 enteros desordenados:

1. **./busq_orden datos3.txt 50 100 200:** El programa ordenará los enteros y luego realizará búsquedas de los números 50, 100 y 200.

```
Windows PowerShell
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos3.txt 50 100 200
Vectores ordenados son iguales.
Datos ordenados:
1 2 2 3 4 5 6 7 8 9 12 12 13 14 15 15 16 17 18 19 20 21 22 24 25 26 26 27 28 29 30 31 31 33 34 35 36 37 38 40 42 43 44 4
5 46 46 47 48 49 50 50 51 52 53 54 55 55 56 57 58 59 60 61 62 62 63 64 65 66 67 68 68 69 69 70 71 71 72 73 74 75 76 77 7
8 79 80 81 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 101 102 103 104 107 108 109 110 110 111 112 115 11
6 118 118 119 119 120 121 121 123 124 125 127 128 128 129 130 130 132 132 132 135 138 139 140 143 143 144 145 145 146 14
8 150
Datos ordenados:
Intercambio -> Tiempo: 0.0906 ms
Burbuja -> Tiempo: 0.0897 ms
Merge -> Tiempo: 0.3019 ms
Buscando 50:
Secuencial en vector original: Encontrado en el indice 66 en 0.0026 ms
Secuencial en vector ordenado: Encontrado en el indice 49 en 0.0039 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 50 en 0.0018 ms
Buscando 100:
Secuencial en vector original: Encontrado en el indice 53 en 0.0019 ms
Secuencial en vector ordenado: Encontrado en el indice 104 en 0.0021 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 104 en 0.0019 ms
Buscando 200:
Secuencial en vector original: NO encontrado en 0.0022 ms
Secuencial en vector ordenado: NO encontrado en 0.0022 ms
Busqueda binaria en vector ordenado: NO encontrado en 0.0018 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos>
```

Si no existe un entero en el archivo, mostrará que no fue posible encontrarlo, como en el caso del número 200 en el ejemplo anterior.

2. **./busq_orden datos3.txt 50**: El programa ordenará los enteros y luego realizará la búsqueda del número 50.

```
Windows PowerShell
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> ./busq_orden datos3.txt 50
Vectores ordenados son iguales.
Datos ordenados:
1 2 3 4 5 6 7 8 9 12 12 13 14 15 15 16 17 18 19 20 21 22 24 25 26 26 27 28 29 30 31 31 33 34 35 36 37 38 40 42 43 44 4
5 46 46 47 48 49 50 50 51 52 53 54 55 55 56 57 58 59 60 61 62 62 63 64 65 66 67 68 68 69 69 70 71 71 72 73 74 75 76 77 7
8 79 80 81 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 101 102 103 104 107 108 109 110 110 111 112 115 11
6 118 118 119 119 120 121 121 123 124 125 127 128 128 129 130 130 132 132 132 135 138 139 140 143 143 144 145 145 146 14
8 150
Datos ordenados:
Intercambio -> Tiempo: 0.0909 ms
Burbuja -> Tiempo: 0.0898 ms
Merge -> Tiempo: 0.3039 ms
Buscando 50:
Secuencial en vector original: Encontrado en el indice 66 en 0.0024 ms
Secuencial en vector ordenado: Encontrado en el indice 49 en 0.0029 ms
Busqueda binaria en vector ordenado: Encontrado en el indice 50 en 0.0019 ms
PS C:\Users\daeld\OneDrive\Documentos\Programas\algoritmos> |
```

- Análisis y Reflexión:

Algorítmicamente hablando, la eficiencia depende en gran medida de la cantidad de elementos en el vector y del número de búsquedas que se realizarán.

- Si solo se va a realizar una búsqueda en un vector grande, podría ser más eficiente buscar en el vector desordenado utilizando la búsqueda secuencial ($O(n)$) en lugar de ordenar el vector primero ($O(n^2)$ o $O(n \log n)$) y luego realizar una búsqueda binaria ($O(\log n)$).
- Sin embargo, si se planea realizar múltiples búsquedas en el mismo conjunto de datos, es más eficiente ordenar el vector una vez ($O(n \log n)$) y luego realizar búsquedas binarias repetidas ($O(k \log n)$).

Por lo que, la elección entre buscar en un vector desordenado o primero ordenar el vector y luego buscar en él, depende completamente de la cantidad de búsquedas que se realizarán y del tamaño del conjunto de datos. Para búsquedas únicas en conjuntos de datos pequeños, la búsqueda secuencial en el vector desordenado puede ser más eficiente. Sin embargo, para conjuntos de datos más grandes o múltiples búsquedas, ordenar primero y luego realizar búsquedas binarias suele ser más eficiente en términos de complejidad algorítmica.