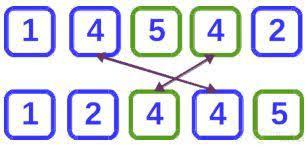
**Intercambio**

El algoritmo de intercambio (Exchange sort en inglés) funciona intercambiando elementos adyacentes en la lista si están en orden incorrecto (Cormen et al., 2022). Este proceso se repite hasta que no requiere realizar más intercambios dando a entender que la lista está ordenada

**Figura 1**



*funcionamiento del algoritmo de intercambio*

**codificación**

#include <iostream>

**void** **intercambio**(**int** arr[], **int** n) {

**for** (**int** i = **0**; i < n - **1**; i++) {

**for** (**int** j = **0**; j < n - i - **1**; j++) {

            // Compara elementos adyacentes

**if** (arr[j] > arr[j + **1**]) {

                // Intercambia si están en el orden incorrecto

                std::swap(arr[j], arr[j + **1**]);

            }

        }

    }

}

**int** **main**() {

**int** arreglo[] = {**64**, **34**, **25**, **12**, **22**, **11**, **90**};

**int** n = **sizeof**(arreglo) / **sizeof**(arreglo[**0**]);

    std::cout << "Arreglo original: ";

**for** (**int** i = **0**; i < n; i++) {

        std::cout << arreglo[i] << " ";

    }

    // Llamada a la función de intercambio

    intercambio(arreglo, n);

    std::cout << "**\n**Arreglo ordenado: ";

**for** (**int** i = **0**; i < n; i++) {

        std::cout << arreglo[i] << " ";

    }

**return** **0**;

}

**Ejecución**

**Texto

Descripción generada automáticamente**

**Análisis**

Función “***Intercambio”***

* La función **intercambio** toma un arreglo **arr** y su longitud **n** como parámetros.
* Utiliza dos bucles anidados para comparar y, si es necesario, intercambiar elementos adyacentes.
* El bucle externo **(for (int i = 0; i < n - 1; i++))** controla el número total de pasadas a través del arreglo.
* El bucle interno **(for (int j = 0; j < n - i - 1; j++))** compara y, si es necesario, intercambia elementos.
* La condición **if (arr[j] > arr[j + 1])** verifica si el elemento actual es mayor que el siguiente.
* Si la condición es verdadera**, std::swap(arr[j], arr[j + 1])** intercambia los elementos.

Función principal “**Main**”

* Se declara un arreglo de números desordenados llamado arreglo.
* La variable n se calcula como la longitud del arreglo.
* Se imprime el arreglo original utilizando un bucle for.
* Se llama a la función intercambio para ordenar el arreglo.
* Se imprime el arreglo ordenado utilizando otro bucle for.

**Conclusiones**

En resumen, el algoritmo de ordenamiento por intercambio, es simple pero menos eficiente en comparación con otros algoritmos más avanzados. Funciona intercambiando repetidamente elementos adyacentes hasta que el arreglo esté ordenado. Aunque es fácil de entender e implementar, su rendimiento no es óptimo, especialmente en conjuntos de datos grandes.