



Centro de enseñanza técnica industrial

Ingeniería en Desarrollo de Software

Estructura de Datos y Algoritmia

Profesor: NAVARRO BUSTOS VICTOR MANUEL

Tarea: Método Burbuja, selección, inserción.

Alumno: Eric Rodrigo Arana Martínez **Reg.** 22310401

Grupo: 4P

Fecha entrega: 19/02/2024

Ordenamiento Burbuja

El ordenamiento burbuja hace múltiples recorridos a lo largo de una lista, donde compara los valores adyacentes e intercambia los que no estén en orden. Cada pasada a lo largo de la lista ubica el siguiente valor más grande en su lugar apropiado. En esencia, cada ítem “burbujea” hasta el lugar al que pertenece.

Complejidad computacional:

Mejor caso: $O(n)$, Caso promedio: $O(n^2)$, Peor caso: $O(n^2)$

```
package burbuja;

public class Burbuja {

    public static void main(String[] args) {
        int[] lista = {32,121,2,65,43};

        System.out.println("Arreglo:");
        imprimir(lista);

        ordenar(lista);

        System.out.println("Ordenado");
        imprimir(lista);
    }

    static void ordenar(int[] lista){
        int n = lista.length;
        for(int i = 0;i<n-1;i++){
            for(int j = 0;j<n-1;j++){
                if(lista[j]>lista[j+1]){
                    int tempo = lista[j];
                    lista[j] = lista[j+1];
                    lista[j+1] = tempo;
                }
            }
        }
    }

    static void imprimir(int[] lista){
        for(int i = 0;i<lista.length;i++){
            System.out.print(lista[i] + " ");
        }
        System.out.println();
    }
}
```

Ordenación por Selección

El método de ordenación por selección consiste en repetir los siguientes pasos: Se busca el elemento más pequeño del array y se coloca en la primera posición. Entre los restantes, se busca el elemento más pequeño y se coloca en la segunda posición. Entre los restantes se busca el elemento más pequeño y se coloca en la tercera posición.

```
package selection;

public class Selection {

    public static void main(String[] args) {
        int[] lista = {87, 32, 121, 2, 65, 43};

        System.out.println("Arreglo:");
        imprimir(lista);

        ordenar(lista);

        System.out.println("Ordenado");
        imprimir(lista);
    }

    static void ordenar(int[] lista) {
        int n = lista.length;
        for (int i = 0; i < n - 1; i++) {
            int minimo = i;
            for (int j = i + 1; j < n; j++) {
                if (lista[j] < lista[minimo]) {
                    minimo = j;
                }
            }

            int tempo = lista[minimo];
            lista[minimo] = lista[i];
            lista[i] = tempo;
        }
    }

    static void imprimir(int[] lista) {
        for (int i = 0; i < lista.length; i++) {
            System.out.print(lista[i] + " ");
        }
        System.out.println();
    }
}
```

Ordenación por Inserción Directa

El método de ordenación por inserción directa consiste en recorrer todo el array comenzando desde el segundo elemento hasta el final. Para cada elemento, se trata de colocarlo en el lugar correcto entre todos los elementos anteriores a él o sea entre los elementos a su izquierda en el array. Dada una posición actual p , el algoritmo se basa en que los elementos $A[0]$, $A[1]$, ..., $A[p-1]$ ya están ordenados.

```
package insert;
public class Insert {
    public static void main(String[] args) {
        int[] lista = {87, 32, 121, 2, 65, 43};

        System.out.println("Arreglo:");
        imprimir(lista);

        ordenar(lista);

        System.out.println("Ordenado");
        imprimir(lista);
    }
    static void ordenar(int[] lista) {
        int n = lista.length;
        for (int i = 1; i < n ; i++) {
            int actual= lista[i];
            int j = i-1;
            while(j>=0 && lista[j]> actual){
                lista[j+1] = lista[j];
                j=j-1;
            }
            lista[j+1] = actual;
        }
    }
    static void imprimir(int[] lista) {
        for (int i = 0; i < lista.length; i++) {
            System.out.print(lista[i] + " ");
        }
        System.out.println();
    }
}
```

Capturas de Pantalla

<pre>package burbuja; public class Burbuja { public static void main(String[] args) { int[] lista = {32,121,2,65,43}; System.out.println("Arreglo:"); imprimir(lista); ordenar(lista); System.out.println("Ordenado"); imprimir(lista); } static void ordenar(int[] lista){ int n = lista.length; for(int i = 0; i < n-1; i++){ for(int j = 0; j < n-1-i; j++){ if(lista[j] > lista[j+1]){ int tempo = lista[j]; lista[j] = lista[j+1]; lista[j+1] = tempo; } } } } static void imprimir(int[] lista){ for(int i = 0; i < lista.length; i++){ System.out.print(lista[i] + " "); } System.out.println(); } }</pre>	<pre>public static void main(String[] args) { int[] lista = {87, 32, 121, 2, 65, 43}; System.out.println("Arreglo:"); imprimir(lista); ordenar(lista); System.out.println("Ordenado"); imprimir(lista); } static void ordenar(int[] lista) { int n = lista.length; for (int i = 0; i < n - 1; i++) { int minimo = i; for (int j = i + 1; j < n; j++) { if (lista[j] < lista[minimo]) { minimo = j; } } int tempo = lista[minimo]; lista[minimo] = lista[i]; lista[i] = tempo; } } static void imprimir(int[] lista) { for (int i = 0; i < lista.length; i++) { System.out.print(lista[i] + " "); } System.out.println(); }</pre>	<pre>public static void main(String[] args) { int[] lista = {87, 32, 121, 2, 65, 43}; System.out.println("Arreglo:"); imprimir(lista); ordenar(lista); System.out.println("Ordenado"); imprimir(lista); } static void ordenar(int[] lista) { int n = lista.length; for (int i = 1; i < n; i++) { int actual = lista[i]; int j = i-1; while (j >= 0 && lista[j] > actual) { lista[j+1] = lista[j]; j = j-1; } lista[j+1] = actual; } } static void imprimir(int[] lista) { for (int i = 0; i < lista.length; i++) { System.out.print(lista[i] + " "); } System.out.println(); }</pre>
<pre>run: Arreglo: 32 121 2 65 43 Ordenado 2 32 43 65 121 BUILD SUCCESSFUL (total time: 0 seconds)</pre>	<pre>run: Arreglo: 87 32 121 2 65 43 Ordenado 2 32 43 65 87 121 BUILD SUCCESSFUL (total time: 0 seconds)</pre>	<pre>run: Arreglo: 87 32 121 2 65 43 Ordenado 2 32 43 65 87 121 BUILD SUCCESSFUL (total time: 0 seconds)</pre>

Referencias:

Algoritmos de ordenación en Java. Método de Ordenación por Selección. (s. f.).

<https://puntocomnoesunlenguaje.blogspot.com/2012/12/java-metodo-ordenacion-seleccion.html>

Ordenamiento burbuja. (s. f.). Página Jimdo de Tutospoo.

<https://tutospoo.jimdofree.com/tutoriales-java/m%C3%A9todos-de-ordenaci%C3%B3n/burbuja/>

Ordenamiento por Inserción directa. (s. f.).

<https://puntocomnoesunlenguaje.blogspot.com/2015/02/ordenamiento-insercion-directa-java.html>