ArgentinaPrograma
YoPrograma

Algoritmos de Ordenamiento

por Leonardo Blautzik, Federico Gasior y Lucas Videla

Julio / Diciembre 2021

Ministerio de
Desarrollo Productivo
Argentina

Ordenamiento

Definición:

El proceso de ordenamiento consiste en disponer un grupo de elementos de acuerdo a algún orden lineal definido sobre los mismos.

Orden lineal

Un orden lineal es una relación que se establece entre los miembros de un conjunto.

Clave (key)

Parte del elemento que es utilizado para determinar el orden lineal del mismo.

Definición

Dado los elementos
e1 , e2 , ..., en
con valores de claves
k1 , k2 , ..., kn
respectivamente, debe resultar el mismo grupo de elementos en orden
ei1, ei2 , ..., ein
tal que
ki1 <= ki2 <= ... <= kin

Características

- No es necesario que todos los elementos tengan distintos valores de clave.
- No es necesario que los elementos con igual valor de clave aparezcan en un orden particular.

Algoritmos de ordenamiento

El objetivo es ordenar un grupo de elementos de tal forma que el valor de sus claves formen una secuencia no decreciente. Existen múltiples algoritmos que son utilizados dependiendo del contexto.

Clasificación

- Internos:
  - Procesos de ordenamiento llevados a cabo en memoria principal.
  - Aprovecha la capacidad de acceso aleatorio en distintas formas.
- Externos:
  - Procesos de ordenamiento llevados a cabo en memoria secundaria.
  - Necesarios cuando el número de elementos a ordenar es demasiado grande para caber en memoria principal.

Propiedades

- **Estabilidad:** Cuando hay elementos con claves repetidas, se mantendrá el orden previo de estos elementos.

{4,5,7,2,9,7',2',5',6,7'',0}
{0,2,2',4,5,5',6,7,7',7'',9}

- **Sensibilidad:** El tiempo de respuesta depende de las características de la entrada.
  - Ordenada o casi ordenada.
  - Orden inverso.
  - Orden aleatorio.

Algoritmon de Ordenamientos Elementales

Son los que analizaremos en el curso estudiando sus características y su función O asociada.

- **Selección**
- **Burbujeo**
- **Inserción**

Algoritmos avanzados (fuera del alcance del curso)

- **Quicksort** $O(n * log(n))$
- **Heapsort** $O(n * log(n))$
- **Shellsort** $O(n^{3/2})$
- **Mergesort** $O(n * log(n))$

Consideraciones

Para explicar los distintos algoritmos de ordenamiento consideraremos el problema de ordenar un vector de número enteros.

Selección

Estrategia:

Poner el mas grande al final en cada iteracion:

En cada iteración se encuentra el máximo de un vector con n elementos y se lleva el máximo a la última posición. Antes de comenzar la nueva iteración, se considera que el tamaño del vector se redujo en 1. Luego de i-ésima iteración los últimos i elementos del vector original están ordenados.

Poner el más chico al principio en cada iteración:

En cada iteración se encuentra el mínimo de un vector con n elementos y se lleva el mínimo a la primera posición. Antes de comenzar la nueva iteración, se considera que el tamaño del vector se redujo en 1. Luego de i-ésima iteración los primeros i elementos del vector original están ordenados.

Visualicemos las estrategias en: https://visualgo.net/en/sorting

En Java

```java
public static void seleccion(Integer[] arreglo) {
    int posMax;
    Integer aux;
    for (int i = arreglo.length - 1; i > 0; i--) {
        posMax = 0;
        for (int j = 0; j <= i; j++) {
            if (arreglo[j] > arreglo[posMax])
                posMax = j;
        }
        aux = arreglo[i];
        arreglo[i] = arreglo[posMax];
        arreglo[posMax] = aux;
    }
}
```

Burbujeo

Estrategia:

Los mas chicos suben:

Imaginar que los elementos a ordenar están almacenados en un arreglo vertical. Los elementos con claves menores son “livianos” y suben. Se recorre varias veces el arreglo de abajo hacia arriba, y al hacer esto, si hay dos elementos adyacentes que no están en orden se invierten.

Los mas grandes bajan:

Imaginar que los elementos a ordenar están almacenados en un arreglo vertical. Los elementos con claves mayores son “pesados” y bajan. Se recorre varias veces el arreglo de arriba hacia abajo, y al hacer esto, si hay dos elementos adyacentes que no están en orden se invierten.

Visualicemos las estrategias en: https://visualgo.net/en/sorting

En Java:

```java
public static void burbujeo(Integer[] arreglo) {
    Integer aux;
    for (int i = 0; i < arreglo.length - 1; i++) {
        for (int j = arreglo.length - 1; j > i; j--) {
            if (arreglo[j] < arreglo[j - 1]) {
                aux = arreglo[j];
                arreglo[j] = arreglo[j - 1];
                arreglo[j - 1] = aux;
            }
        }
    }
}
```

Mejorado

```java
public static void burbujeoW(Integer[] arreglo) {
    Integer aux; boolean hayIntercambio = true; int i = 0;
    while(hayIntercambio && i < arreglo.length - 1) {
        hayIntercambio = false;
        for (int j = arreglo.length - 1; j > i; j--) {
            if (arreglo[j] < arreglo[j - 1]) {
                hayIntercambio = true;
                aux = arreglo[j];
                arreglo[j] = arreglo[j - 1];
                arreglo[j - 1] = aux;
            }
        }
        i++;
    }
}
```

Insersión

Estrategia:

Considera una parte del vector ya ordenado e inserta cada uno de los elementos restantes en el lugar que le corresponde. En el i-ésimo recorrido se “inserta” el i-ésimo elemento en el lugar correcto.

Visualicemos la estrategia en: https://visualgo.net/en/sorting

En Java

```java
public static void insercion(Integer arreglo[]) {
    int i, j;
    Integer aux;
    for (i = 1; i < arreglo.length; i++) {
        aux = arreglo[i];
        j = i - 1;
        while ((j >= 0) && (arreglo[j] > aux)) {
            arreglo[j+1] = arreglo[j];
            j--;
        }
        arreglo[j+1] = aux;
    }
}
```

¡Muchas Gracias!

continuará...

Ministerio de
Desarrollo Productivo
Argentina