

INF221 – Algoritmos y Complejidad

Clase #18 Complejidad

Aldo Berrios Valenzuela

Martes 11 de octubre de 2016

1. Métodos simples de ordenamiento

Algunos ejemplos:

- Bubble Sort.
- Insertion Sort.
- Selection Sort.

Observación: Decimos que son métodos “simples” de ordenamiento porque no son recursivos.

1.0.1. Bubble Sort

Sea a un arreglo, donde $a[i]$ corresponde al i -ésimo elemento de a . El algoritmo consiste en intercambiar $a[i]$ con $a[i+1]$ si están fuera de orden. Esto podemos lograrlo vía dos for anidados, $0 \leq i < n$ y $0 \leq j < i$, tal como muestra el listado 1.

Listado 1: Método de la burbuja

```
void
sort(double a[], int n)
{
    int k;
    for (int i = n - 1; i; i = k) {
        k = 0;
        for (int j = 0; j < i; j++) {
            if (a[j + 1] < a[j]) {
                double tmp;
                tmp = a[j]; a[j] = a[j + 1]; a[j + 1] = tmp;
                k = j;
            }
        }
        i = k;
    }
}
```

Analizamos el mejor y peor de este algoritmo:

- *Mejor caso* – Los elementos de a ya se encuentran ordenados, por lo tanto, se hacen 0 asignaciones (no hay intercambios).

- *Peor caso* – Los elementos de a están todos al revés. Por lo tanto, tendríamos que hacer asignaciones para cada $a[i]$. Esto se traduce a:

$$\sum_{1 \leq i \leq n} (n - i) = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$$

Por otro lado, una aproximación asintótica de las asignaciones que se hacen son:

$$\frac{3n^2}{2} \quad (1.1)$$

Note que el 3 que viene de (1.1) es debido a que en cada iteración tenemos que hacer 3 asignaciones.

1.0.2. Insertion Sort

Definimos el arreglo a , donde $a[i]$ corresponde al i -ésimo elemento de a . Básicamente, este algoritmo en cada una de sus iteraciones divide a en dos porciones: primero los elementos ordenados y luego los elementos que no están ordenados (figura 1). Luego, cuando selection sort encuentra un elemento $a[i]$ tal que $a[i-1] > a[i]$, el

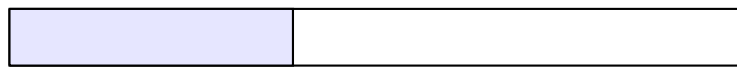


Figura 1: La parte izquierda del arreglo a (la que está pintada) ya se encuentra ordenada. Todo lo que venga después se encuentra desordenado.

algoritmo traslada una porción del sector ya ordenado tal que $a[i]$ quedará en una posición j siempre y cuando $a[j-1] < a[i] < a[j+1]$, con $i > j$. La figura 2 muestra esta operación. Puede observar el detalle de este algoritmo

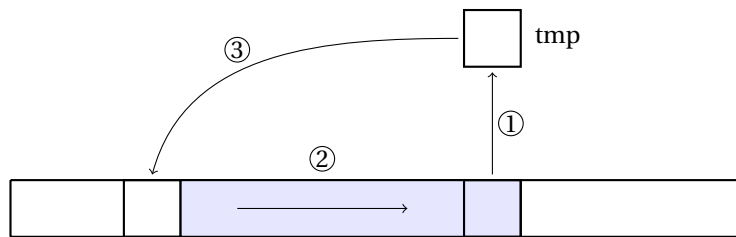


Figura 2: Operación del método de inserción. Importante destacar que el área pintada ya está ordenada.

mirando el listado 2.

Listado 2: Método de la inserción

```
void
sort(double a[], int n)
{
    for (int i = 1; i < n; i++) {
        double tmp = a[i];
        int j;
        for (j = i - 1; j >= 0 && tmp < a[j]; j--)
            a[j + 1] = a[j];
        a[j + 1] = tmp;
    }
}
```

Realizamos el análisis correspondiente para el mejor y peor caso:

- *Mejor caso* – Todos los elementos de a se encuentran ya ordenados, y entonces, este algoritmo hace $2n$ asignaciones.
- *Peor caso* – En este caso, se tiene que:

$$\frac{n^2}{2}$$

1.0.3. Selection Sort

El listado 3 describe el algoritmo de selection sort. Dejamos el análisis para la casa.

Listado 3: Método de selección

```
void
sort(double a[], int n)
{
    for(int i = 0; i < n; i++) {
        int imin = i; double min = a[i];
        for(int j = i + 1; j < n; j++)
            if(a[j] < min) {
                imin = j; min = a[j];
            }
        double tmp = a[i]; a[i] = a[imin]; a[imin] = tmp;
    }
}
```

1.1. Caso Promedio

Requiere saber la distribución de los datos de entrada. Es mucho pedir, en general... usar algún modelo “razonable” (simple de manejar). Para ordenamiento, resulta útil suponer que todos los elementos son diferentes, y que todos los posibles órdenes de datos son igualmente probables.

Como los métodos comparan elementos (los valores no interesan, sólo el orden relativo), basta considerar permutaciones $1, \dots, n$.

Definición 1.1. En una permutación π , hay una *inversión* si $\pi(i) > \pi(j)$ para $i < j$.

Ejemplo 1.1 (Inversiones). Supongamos que tenemos el siguiente arreglo:

$$[2, 4, 1, 3, 5] \tag{1.2}$$

Como podemos observar, entre el 4 y el 1 hay una inversión, porque el 4 está antes que el 1. Cumple con la definición de inversión.

Si aplicamos el método de la burbuja sobre el arreglo (1.2), vemos que con cada intercambio que hace este algoritmo va eliminando una inversión del arreglo:

$$[2, 4, 1, 3, 5] \rightarrow [2, 1, 4, 3, 5]$$

$$[2, 1, 4, 3, 5] \rightarrow [1, 2, 4, 3, 5]$$

$$[1, 2, 4, 3, 5] \rightarrow [1, 2, 3, 4, 5]$$

por lo tanto, se concluye que el número de inversiones que hace el método de la burbuja es igual al número de intercambios que realiza. ■

Es central el número promedio de inversiones.

1.2. Funciones generatrices cumulativas

Definición 1.2 (Objetos rotulados). Son aquellos que les colocamos etiquetas que nos permiten identificar los elementos de una estructura de datos. Por ejemplo, las etiquetas de un nodo en un grafo cualquiera nos permite diferenciarlo entre los demás nodos.

Definición 1.3 (Objetos no rotulados). Son todos aquellos objetos que no tienen etiquetas para identificarlos dentro de una estructura de datos. Por ejemplo, si a los nodos de un grafo no les colocamos estas etiquetas, intercambiar dos nodos entre sí al final del día no produce cambios, porque no tenemos la forma de cómo identificar si realmente se hizo algún cambio.

Algunas cosas importantes que destacar sobre objetos rotulados y no rotulados:

- Cuando la estructura de datos es rotulada, usamos funciones generatrices exponenciales.
- Cuando la estructura de datos es no rotulada, usaremos funciones generatrices ordinarias.

Idea: Tengo una clase de objetos \mathcal{A} (rotulado/no rotulado), con función generatriz:

$$\hat{\mathcal{A}}(z) = \sum_{\alpha \in \mathcal{A}} \frac{z^{|\alpha|}}{|\alpha|!}$$

Para este caso, se tiene que \mathcal{A} es el conjunto que tiene todas las permutaciones del arreglo a , y α es una de esas permutaciones.

Nos interesa la característica $\chi(\alpha)$, que en nuestro caso corresponde al número de inversiones de la permutación α (en caso de un árbol, puede ser la altura del árbol), con lo que tiene sentido definir:

$$\hat{X}(z) = \sum_{\alpha \in \mathcal{A}} \chi(\alpha) \frac{z^{|\alpha|}}{|\alpha|!} \quad (1.3)$$

Es importante destacar, que la sumatoria de (1.3) representa a todas las características de \mathcal{A} .

Continuamos con el desarrollo de (1.3):

$$[z^n] \hat{X}(z) = \frac{1}{n!} \sum_{|\alpha|=n} \chi(\alpha)$$

Luego, el número promedio de características viene dado por:

$$\bar{\chi}_n = \frac{[z^n] \hat{X}(z)}{[z^n] \hat{A}(z)} \quad (1.4)$$

En nuestro caso, sea $\iota(\pi)$ el número de inversiones en la permutación π . Sabemos que podemos describir la clase de permutaciones recursivamente como:

$$P = E + P \star Z$$

Donde:

$$A \star B = \{(\alpha, \beta) : \alpha \in A, \beta \in B; \text{rótulos reasignados en forma consistente}\}$$

Por el método simbólico:

$$\begin{aligned} \hat{P}(z) &= 1 + z\hat{P}(z) \\ \hat{P}(z) &= \frac{1}{1-z} \\ [z^n] \hat{P}(z) &= 1 \rightsquigarrow \frac{p_n}{n!}, p_n = n! \end{aligned}$$

Donde p_n es el número de permutaciones de tamaño n .

Para $\iota(\pi)$, nos interesa la suma sobre *todas* las permutaciones que resultan de $\pi \star (1)$.

Las permutaciones resultantes tienen las mismas inversiones que π más las inversiones en que participa el último elemento agregado.

El total de inversiones en este conjunto:

$$\begin{aligned} &\underbrace{(|\pi| + 1) \iota(\pi)}_{\text{Aporte de } \pi} + \underbrace{\sum_{1 \leq j \leq |\pi|+1} (|\pi| + 1 - j)}_{\text{Aporte de (1) según el rótulo } j} \\ &+ \frac{|\pi| (|\pi| + 1)}{2} \end{aligned}$$

De la descomposición:

$$\sum_{\pi \in P} \iota(\pi) \frac{z^{|\pi|}}{|\pi|!} = \iota(e) + \sum_{\pi \in P} \left((|\pi| + 1) \iota(\pi) + \frac{|\pi| (|\pi| + 1)}{2} \right) \frac{z^{|\pi|+1}}{(|\pi| + 1)!}$$

Y luego:

$$\begin{aligned} \hat{I}(z) &= 0 + \sum_{\pi \in P} (|\pi| + 1) \iota(\pi) \frac{z^{|\pi|+1}}{(|\pi| + 1)!} + \sum_{\pi \in P} \frac{|\pi| (|\pi| + 1)}{2} \frac{z^{|\pi|+1}}{(|\pi| + 1)!} \\ &= z \hat{I}(z) + z \sum_{\pi \in P} \frac{|\pi|}{2} \cdot \frac{z^{|\pi|}}{|\pi|!} + z \sum_{n \geq 0} \frac{n}{2} z^n \\ \hat{I}(z) &= \frac{1}{2} \frac{z^2}{(1-z)^3} \\ [z^n] \hat{I}(z) &= \frac{n(n-1)}{4} \end{aligned}$$