Propiedades de Insertion Sort

Insertion Sort es un algoritmo correcto

¿Qué propiedades debe tener para ser correcto?

Correctitud de Insertion Sort

Para que un algoritmo sea correcto, debe:

- Terminar su ejecución
- Cumplir su objetivo

En este caso el objetivo es ordenar

Propiedades de Insertion Sort

¿Cómo demostramos que el algoritmo termina?

¿Cómo demostramos que ordena?

El algoritmo *Insertion Sort*

Para la secuencia inicial de datos, A:

1. Definir una secuencia ordenada, B, inicialmente vacía
2. Tomar el primer dato x de A y sacarlo de A
3. Insertar x en B de manera que B quede ordenado
4. Si quedan elementos en A, volver a 2.

Finitud

En cada paso se saca un elemento de A y se inserta en B

Cuando no quedan elementos en A, el algoritmo termina

La inserción requiere como máximo recorrer todo B

Como A y B son finitos, el algoritmo termina en tiempo finito

Corrección

PD: Al terminar la n-ésima iteración, B se encuentra ordenada

Podemos hacerlo por **inducción** sobre i

Caso Base: Después de la primera iteración, B tiene un solo dato

$\rightarrow B$ está ordenada

Hipótesis Inductiva: Después de la i-ésima iteración**,** B está ordenada

Demostraremos que después de la iteración $i + 1$, B está ordenada

Extraemos el primer elemento de A, y lo insertamos ordenadamente en B. Termina el paso $i + 1$ y B tiene $i \ + \ 1$ elementos ordenados

En particular, al terminar el algoritmo después del paso n, B está ordenada.

Complejidad de Insertion Sort

Vimos la clase pasada que Insertion Sort es $O(n^2)$

Pero, ¿Qué tiempo toma si los datos vienen ordenados?

$$\boldsymbol{insertionSort}(\boldsymbol{A}, \boldsymbol{n}):$$

$$\boldsymbol{for}\ \boldsymbol{i} = \mathbf{1} \mathinner{\ldotp\ldotp} \boldsymbol{n} - \mathbf{1}:$$

$$\boldsymbol{j} = \boldsymbol{i}$$

$$\boldsymbol{while}\ (\boldsymbol{j} > \mathbf{0}) \wedge (\boldsymbol{A}[\boldsymbol{j}] < \boldsymbol{A}[\boldsymbol{j} - \mathbf{1}]):$$

Intercambiar $\boldsymbol{A}[\boldsymbol{j}]$ con $\boldsymbol{A}[\boldsymbol{j} - \mathbf{1}]$

$$\boldsymbol{j} = \boldsymbol{j} - \mathbf{1}$$

Complejidad de Insertion Sort

Parecería que la complejidad de insertion sort depende de que tan ordenados vienen los datos.

¿Cómo podemos medir esto?

Inversiones

Sea A un arreglo con n números distintos de 1 a n

Si $i < j$ pero $\mathrm{A[i]} > \mathrm{A[j]}$, se dice que (i, j) son un **par invertido**

El número de pares invertidos es una métrica de **desorden**

Complejidad de Insertion Sort

Tenemos un arreglo A de largo n que tiene k **inversiones**

¿Cuanto tiempo toma Insertion Sort en ordenar A?

¿Cuántas inversiones se arreglan con un intercambio?

$\boldsymbol{insertionSort(A, n)}$:

$\boldsymbol{for\ i = 1 \,..\, n - 1}$:

$\boldsymbol{j = i}$

$\boldsymbol{while}\ (\boldsymbol{j} > \boldsymbol{0}) \land (\boldsymbol{A}[\boldsymbol{j}] < \boldsymbol{A}[\boldsymbol{j} - \boldsymbol{1}])$:

Intercambiar $\boldsymbol{A}[\boldsymbol{j}]$ con $\boldsymbol{A}[\boldsymbol{j} - \boldsymbol{1}]$

$\boldsymbol{j = j - 1}$

Antes de cada intercambio se hace una comparación, y esos datos se intercambian solo si están invertidos

Por lo tanto, cada intercambio de elementos adyacentes en el arreglo deshace exactamente una inversión

Además, cada elemento se compara al menos una vez

Complejidad de Insertion Sort

La complejidad es entonces $O(n + k)$

¿Qué valor tiene k en el mejor caso? ¿Y el en peor?

¿Qué hay del caso promedio?

Complejidad de Insertion Sort

Sea $K(A)$ el número de pares invertidos en A

Sea $H(A)$ el número de pares no invertidos en A

¿Cuánto es $K(A) + H(A)$ si A tiene n elementos?

La cantidad de pares que existen en un arreglo de n elementos es

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

Cada par puede estar ó no estar invertido, por lo tanto:

$$K(A) + H(A) = \frac{n^2 - n}{2}$$

Dado n, definimos $\mathcal{A}$ como la secuencia ordenada de todos los números del 1 al n.

Sea $p(\mathcal{A})$ una permutación de $\mathcal{A}$. Definimos $P(\mathcal{A})$ como el conjunto de todas las posibles permutaciones de $\mathcal{A}$.

$P(\mathcal{A})$ contiene todos los posibles inputs de largo n que puede recibir un algoritmo de ordenación.

Complejidad de Insertion Sort

Sea $P(\mathcal{A})$ el conjunto de permutaciones de $\mathcal{A}$

Queremos encontrar el número de inversiones promedio:

$$\mathbb{E}\big(K(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|} \sum_{a \in P(\mathcal{A})} K(a)$$

Notar que

$$\mathbb{E}\big(K(\mathcal{A})\big) = \mathbb{E}\big(H(\mathcal{A})\big)$$

Al sumarlos, queda

$$\mathbb{E}\big(K(\mathcal{A})\big) + \mathbb{E}\big(H(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|}\left(\sum_{a \in P(\mathcal{A})} K(a) + \sum_{a \in P(\mathcal{A})} H(a)\right)$$

$$2 \cdot \mathbb{E}\big(K(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|}\left(\sum_{a \in P(\mathcal{A})} K(a) + H(a)\right)$$

$$2 \cdot \mathbb{E}\big(K(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|}\left(\sum_{a \in P(\mathcal{A})} K(a) + H(a)\right)$$

$$2 \cdot \mathbb{E}\big(K(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|}\left(\sum_{a \in P(\mathcal{A})} \frac{n^2 - n}{2}\right)$$

$$2 \cdot \mathbb{E}\big(K(\mathcal{A})\big) = \frac{1}{|P(\mathcal{A})|} \cdot |P(\mathcal{A})| \cdot \frac{n^2 - n}{2}$$

$$\mathbb{E}\big(K(\mathcal{A})\big) = \frac{n^2 - n}{4} \in \Theta(n^2)$$

Complejidad de *Insertion Sort*

La cantidad de inversiones promedio es entonces $O(n^2)$

Eso significa que Insertion Sort es $O(n^2)$ en el caso promedio

Si un algoritmo sólo resuelve una inversión por intercambio, no puede ordenar más rápido que $O(n^2)$ en promedio y por lo tanto en el peor caso