	INSERTION-SORT(A, n)	cost	times
1	**for** $i = 2$ **to** n	c_1	n
2	$key = A[i]$	c_2	$n-1$
3	// Insert $A[i]$ into the sorted subarray $A[1:i-1]$	0	$n-1$
4	$j = i-1$	c_4	$n-1$
5	**while** $j > 0$ and $A[j] > key$	c_5	$\sum_{i=2}^{n} t_i$
6	$A[j+1] = A[j]$	c_6	$\sum_{i=2}^{n} (t_i - 1)$
7	$j = j-1$	c_7	$\sum_{i=2}^{n} (t_i - 1)$
8	$A[j+1] = key$	c_8	$n-1$

Son "n" veces ya que se cuenta la ultima iteración para salir del bucle

$\sum_{i=2}^{n} 1 = n-1 \rightarrow n+1$

t_i : Representa la cantidad de ejecuciones de la linea 5 para una cierta iteración i

1 más para salir del bucle

IMPORTANTE DEFINIR NUESTRAS VARIABLES

El tiempo de ejecución será esa sumatoria ponderada

Que tiene variables que no solamente dependen de "n" sino de "i"

Podemos sacar cual es el peor y mejor caso

1° t_i esta acotado → mínimo hacemos 1 ejecución (ya que leemos el while)

$$1 \le t_i \le i$$

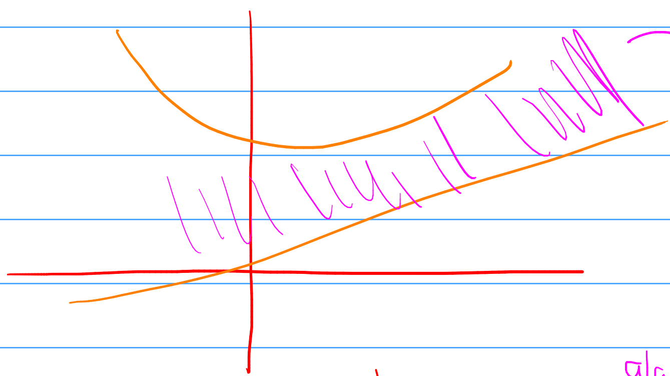
ya que el while retrocede como maximo $i-1$ hasta 0 (salteo de 1 en 1)

2° Sacando calculos podemos concluir que tiempo de ejecución $T(n)$ esta acotado superiormente por alguna expresión cuadratica y una expresión lineal

$$c\,\underbrace{\bigcirc}_{cte} + n\,\underbrace{\bigcirc}_{cte} \le T(n) \le \underbrace{\bigcirc}_{cte} n^2 + \underbrace{\bigcirc}_{cte} n + \underbrace{\bigcirc}_{cte} c$$

Obs: A nosotros no nos interesa ser especificos con estas constantes porque nos queda claro como oscila la función T

T:



Oscila entre alguna cuadratica y lineal

Pero es muy largo escribir todo esto:

$$\boxed{c\,\underbrace{\bigcirc}_{cte} + n\,\underbrace{\bigcirc}_{cte} \le \boxed{T(n)} \le \underbrace{\bigcirc}_{cte} n^2 + \underbrace{\bigcirc}_{cte} n + \underbrace{\bigcirc}_{cte} c}$$

alguna notación que me diga que esta acotado superiormente por una cuadratica

alguna notación que me diga que esta acotado inferiormente por una lineal

Ahora sabemos que esta acotado por esto:

$$T(n) \leq \underbrace{()}_{cte} n^2 + \underbrace{()}_{cte} n + \underbrace{()}_{cte} c$$

Si quisieramos reducirlo a n^2
y quisieramos acotarlo a n^2
Hay un problema

Se puede ver que n^2 es menor a n en cierto intervalo lo cual NO CUMPLE

Ahora la solucion es REESCALAR
Para que ahi si este acotada para un cierto n

Entonces lo que queremos es:

Decir que:

$T(n)$ esta acotado superiormente por un reescalamiento de n^2 a partir de un cierto punto inicial

$$\Longrightarrow \quad T(n) = O(n^2)$$

osea $T(n)$ vive en ese conjunto n^2

son lo mismo

La Notación a esta expresión es O-Notation (Big O)

$O(g(n)) = \{ F(n)$; Existe una constante "C" tal que

$$0 \leq F(n) \leq C g(n) \quad \forall \; n \geq n_0 \}$$

En otras palabras, $O(g(n))$ son todas las funciones $F(n)$ que pueden ser acotadas superiormente por un Reescalar de cierta función $g(n)$ a partir de un cierto punto → Puede ser cualquier escalar y valor inicial)

Con tal que cumpla eso se dice que es Big (O)

son lo mismo

traduccion más reducida:

$$T(n) \leq C n^2 \quad \forall \; n \geq n_0 \; \wedge \; c, n_0 > 0$$

Ejm:

Probar que: $n^2 + 10n + 2 = O(n^2)$

Esto es lo mismo que decir que:

Parafraseo

Probar que: $\exists\ c, n_0 > 0$ tq $n \geq n_0$ $\longrightarrow$ $n^2 + 10n + 2 \leq c n^2$

escalar cierto punto

Probar que $n^2+10n+2$ esta acotado superiormente por un reescalamiento de n^2

Borrador:

Truco: separar los terminos y acotar cada uno

$n^2 \leq n^2$

$10n \leq n^2$, para $\forall\ n \geq 10 \rightarrow n_0$

$2 \leq n^2$, $\forall\ n \geq \sqrt{2}$

$n^2 + 10n + 2 \leq 3n^2$ $\rightarrow c$

Limpio:

Usaremos $c = 3 \wedge n_0 = 10$

Para probar que:

Como: $n \geq 10 \xrightarrow[\text{que}]{\text{probaremos}} \underbrace{n^2 + 10n + 2}_{F(n)} \leq 3n^2$

$n \geq 10$

$n^2 \geq n^2$

$n^2 \geq 10n$

$n^2 \geq 10n \geq 2$

+

$n^2 + 10n + 2 \leq 3n^2$

$\Rightarrow \forall\ n \geq 10 \longrightarrow F(n) \leq 3n^2$

$F(n) = O(n^2)$ $\square$

Ω - Notation

↳ Notación que me permite acotar inferiormente

Es decir

$T(n)$ esta acotado inferiormente por un reescalamiento de $g(n)$

Función

a partir de un instante

$$\Omega(g(n)) = \{ f(n), \text{ existe una constante } c, n_0 \geq 0 \text{ tq } 0 \leq c\, g(n) \leq f(n) \ \forall n \geq n_0 \}$$

Hay una notación que acota inferior y superiormente por escalares de la misma función a partir de un punto inicial

↳ O sea:

$$\exists\, c_1, c_2, n_0 > 0 \text{ tq } \forall n \geq n_0 \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n)$$

Esto se llama Big-theta $\Rightarrow$ Big-Θ

Θ - Notación : Son todas las funciones que pueden ser acotadas tanto por arriba como por abajo de escalares de $g(n)$ a partir de un cierto punto

↳ n_0

$$\Theta(g(n)) = \{ f(n), \text{ donde existen constantes } c_1, c_2, n_0 > 0 \text{ tq } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) ;\ \forall n \geq n_0 \}$$

↳ Esto tambien es lo mismo a decir que:

$$\Theta(g(n)) \equiv O(g(n)) \wedge \Omega(g(n))$$

acotada superiormente por un $c_1 g(n)$

y

acotada inferiormente por $c_2 g(n)$

mismo $g(n)$

Otra forma de definir Θ es que cumpla O y Ω a la vez

Ejm:

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

$\exists\ c_1, c_2, n_0$ tq $\forall\ n \geq n_0 \xrightarrow[\text{que}]{\text{queremos}} c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$

→ En este tipo de problemas primero buscar la mas sencilla

1°

$$\frac{1}{2}n^2 - 3n \leq n^2 \quad ✓ \quad \forall n \geq 1$$

$\therefore\ c_2 = 1$

2°

$c_1 n^2 \leq \frac{1}{2}n^2 - 3n$ ...... (torta: $\frac{1}{2}n^2$)

TRUCO: buscar el mas grande sera mi torta y los demas limitarlos a mitad de torta ya que así al quitarlos lo que tengas sera MAYOR a lo limitado

$$3n \leq \frac{1}{2}\left(\frac{1}{2}n^2\right)$$

$$3n \leq \frac{1}{4}n^2 \longrightarrow \forall\ n \geq 12$$

$$\frac{1}{2}n^2 - 3n \geq \frac{1}{2}n^2 - \frac{1}{2}\left(\frac{1}{2}n^2\right) = \frac{1}{4}n^2$$

$$\frac{1}{2}n^2 - 3n \geq \frac{1}{4}n^2$$

$\frac{1}{4} \longrightarrow c_1$

Con $c_1 = \frac{1}{4}$, $c_2 = 1$, $n_0 = 12$

$$\therefore\ \frac{1}{4}n^2 \leq \frac{1}{2}n^2 - 3n \leq 1n^2, \quad \forall\ n \geq 12$$

$$\therefore\ \frac{1}{2}n^2 - 3n = \Theta(n^2) \quad \text{D}$$