| | 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

$\sum_{i=2}^{n} 1 = n-1$ → $n+1$

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

IMPORTANTE DEFINIR NUESTRAS VARIABLES

1 más para salir del bucle

El tiempo de ejecución sera 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)

$$1 \le t_i \le i$$

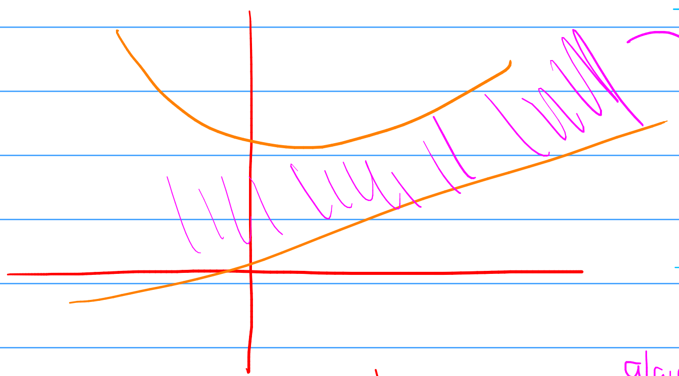
ya que el while retrocede como maximo $i-1$ hasta 0 (salteo de 1 en 1)

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 específicos 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:

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

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

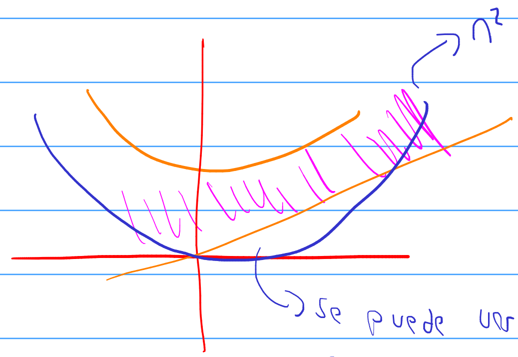
alguna notación que me diga que esta acotado inferiormente por una lineal

Ahora sabemos que esta acotado por esto:

$$T(n) \leq \bigcirc n^2 + \bigcirc n + \bigcirc c$$

cte cte cte

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 solución es REESCALAR
Para que ahi si este acotada para un cierto $n$

Od

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

son lo mismo

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

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

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

$$O(g(n)) = \{ F(n) : \text{Existe una constante "c" tal que } 0 \leq F(n) \leq c\,g(n) \ \forall \ n \geq n_0 \}$$

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

son lo mismo

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

traduccion más reducida:

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

Ejm:

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

Esto es lo mismo que decir que:

Parafrasio

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 $\forall\ n \geq 10 \to n_0$

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

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

$3 \to c$

Limpio:

Usaremos $c = 3 \wedge n_0 = 10$

Para probar que:

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

$n \geq 10$

$n^2 \geq n^2$ ✓

$n^2 \geq 10n$

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

+

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

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

$F(n) = O(n^2)$ □