Desarrollo Guía 1 – Complejidad de Algoritmos

Conceptos básicos

$f(n) = \mathrm{O}\big(g(n)\big)$, $g(n)$ es cota Asintótica Superior de $f(n)$

Se demuestra:

$$\mathrm{O}\big(g(n)\big) = \begin{Bmatrix} f: N \to R^+ \mid \exists\, c > 0 \wedge n_0 \in N: \\ f(n) \le \mathrm{c} * \mathrm{g}(n),\ \forall\, n \ge n_0 \end{Bmatrix}$$

Spoiler:

$f(n) = \Omega\big(g(n)\big)$, $g(n)$ es cota Asintótica Inferior de $f(n)$

$f(n) = \Theta\big(g(n)\big)$, $g(n)$ es cota Asintótica Superior e Inferior de $f(n)$

Desarrollo

1. Obtener el $g(n)$, c y n_0

(a) $f(n) = 2n + 5$

Queremos demostrar:
$2n + 5 \ \in \ \mathrm{O}(n)$

Por lo tanto:
$2n + 5 \ \le \ \mathrm{c} * \mathrm{n}$
$5 \ \le \ \mathrm{cn} - \ 2n$
$5 \ \le \ (\mathrm{c} - 2)n$

$(n = 1) \to$
$5 \ \le \ \mathrm{c} - 2 \quad /{+2}$
$7 \le \ \mathrm{c}$

Si queremos que nuestro n_0 sea lo más pequeño posible AKA 1, debemos asegurar que $5 \ \le \ (\mathrm{c} - 2)n$ se cumpla pata todos los valores de n mayores a 1, por lo que se reemplazara 1 en n

Finalmente tenemos que:

$2n + 5 \ \le \ 7\mathrm{n} \quad , \forall\, n \ge \ 1$

$g(n) = n$
$c = 7$
$n_0 = 1$

(b) $f(n) = 5n^2 + 4n$

Queremos demostrar:
$5n^2 + 4n \in \mathrm{O}(n^2)$

Por lo tanto:
$5n^2 + 4n \leq cn^2 \quad /\div n$
...
$4 \leq (\mathrm{c} - 5)\mathrm{n}$

$(n = 1) \rightarrow$
$4 \leq \mathrm{c} - 5$
$9 \leq \mathrm{c}$

Finalmente tenemos que:

$5n^2 + 4n \leq 9n^2 \quad , \forall\, n \geq 1$

$g(n) = n^2$
$c = 9$
$n_0 = 1$

(c) $f(n) = 8n^3 + 2n\log(n)$

Queremos demostrar:
$8n^3 + 2n\log(n) \in \mathrm{O}(n^3)$

Por lo tanto:
$8n^3 + 2n\log(n) \leq cn^3 \quad /\div n$
$8n^2 + 2\log(n) \leq cn^2$

$(n = 1) \rightarrow$
$8 + 2(0{,}69) \leq \mathrm{c}$
$9{,}38 \leq \mathrm{c} \rightarrow \mathrm{c} \geq 10$

Finalmente tenemos que:

$8n^3 + 2n\log(n) \leq 10n^3 \quad , \forall\, n \geq 1$

$g(n) = n^3$
$c = 10$
$n_0 = 1$

(d) $f(n) = \frac{n(n-1)}{2} + n^2\sqrt{n}$

Queremos demostrar:

$$\frac{n(n-1)}{2} + n^2\sqrt{n} \in \mathrm{O}\left(n^2\sqrt{n}\right)$$

También se puede expresar como:

$$\frac{n(n-1)}{2} + n^{\frac{5}{2}} \in \mathrm{O}\left(n^{\frac{5}{2}}\right)$$

Por lo tanto:

$$\frac{n(n-1)}{2} + n^{\frac{5}{2}} \leq \mathrm{c}n^{\frac{5}{2}} \quad /* \frac{2}{n}$$

$$n - 1 + 2n^{\frac{3}{2}} \leq 2\mathrm{c}n^{\frac{3}{2}}$$

$$n + 2n^{\frac{3}{2}} - 2\mathrm{c}n^{\frac{3}{2}} \leq 1$$

En este caso se probó con n = 2, dado que al probar n = 1 da resultado c = 1, lo cual solo cumple la condición para n = 1, por lo que no se garantiza que este acotada en todos los puntos mayores a 1

$(n = 2) \rightarrow$

$$2 + 2 * 2^{\frac{3}{2}} - 2 * \mathrm{c} * 2^{\frac{3}{2}} \leq 1$$

$$1 + 2 * 2^{\frac{3}{2}} \leq 2 * \mathrm{c} * 2^{\frac{3}{2}} \quad /2 * 2^{\frac{3}{2}}$$
$$1 + 1 \leq \mathrm{c}$$
$$2 \leq \mathrm{c}$$

Finalmente tenemos que:

$$\frac{n(n-1)}{2} + n^2\sqrt{n} \leq 2n^2\sqrt{n} \quad , \forall\, n \geq 1$$

$g(n) = n^{\frac{5}{2}}\; o\; n^2\sqrt{n}$
$c = 2$
$n_0 = 1$

3. Suponga que el algoritmo MergeSort en el peor de los casos, ordena un conjunto de objetos realizando $t(n) = 64\ n\ log_2(n)$ comparaciones, donde n es el tamaño de la entrada.
   Para que valores de n InsertSort es mejor que MergeSort, en el peor de los casos?

Tenemos
MergeSort: $t(n) = 64\ n\ log_2(n)$
InsertSort: $t(n) = \frac{n^2}{2} - \frac{n}{2}$

Ambos son peor de los casos, en clases usamos

Queremos demostrar
$\frac{n^2}{2} - \frac{n}{2} \leq 64\ n\ log_2(n)\ , \forall\ n_0 \leq n \leq n_1$

Por lo tanto
$$\frac{n^2}{2} - \frac{n}{2} \leq 64\ n\ log_2(n) \quad /\div n$$
$$\frac{n}{2} - \frac{1}{2} \leq 64\ log_2(n) \quad /*\ 2$$
$$n - 1\ \leq 128\ log_2(n) \quad /*\ 2$$
$$n - 128\ log_2(n)\ \leq 1$$

Dada la complejidad del ejercicio la mejor solución es probar valores para acercarnos a la solución

$(n = 1) \rightarrow$
$1 - 128\ log_2(1)\ \leq 1$
Si cumple, por lo que esta será nuestro n_0

$(n = 2000) \rightarrow$
$2000 - 128\ log_2(2000)\ \leq 1$
$2000 - 1403\ \leq 1$
No cumple, es menos

$(n = 1000) \rightarrow$
$1000 - 128\ log_2(1000)\ \leq 1$
$1000 - 1275\ \leq 1$
Cumple, pero es un poco más

…
Varios intentos después
$(n = 1300) \rightarrow$
$1300 - 128\ log_2(1300)\ \leq 1$
$1300 - 1324\ \leq 1$
Cumple, aunque no es exacto, pero próximo
Por lo que 1300 es nuestro n_1

Finalmente tenemos que
$\frac{n^2}{2} - \frac{n}{2} \leq 64\ n\ log_2(n)\ , \forall\ 1 \leq n \leq 1300$
Por lo que se concluye que InsertSort es mejor que MergeSort al dale entre 1 y 1300 elementos.