

181935

Estructuras de Datos Avanzadas

Tarea 1

1. Para cada función $f(n)$ y tiempo t determine el tamaño máximo del problema (la n) que puede resolverse en tiempo t . Suponga que el algoritmo usado para resolver el problema toma $f(n)$ microsegundos (reporte sólo el orden de magnitud si los números son demasiado grandes)

	1 Segundo	1 Minuto	1 Hora	1 Día	1 Mes	1 Año	1 Siglo
$\log_2(n)$	$2^{1 \times 10^6}$	$2^{6 \times 10^7}$	$2^{3.6 \times 10^9}$	$2^{8.6 \times 10^{10}}$	$2^{2.62 \times 10^{12}}$	$2^{3 \times 10^{13}}$	$2^{3.14 \times 10^{15}}$
\sqrt{n}	1×10^{12}	6×10^{14}	3.6×10^{18}	8.64×10^{20}	2.62×10^{24}	3.15×10^{26}	3.15×10^{30}
N	1×10^6	6×10^7	3.6×10^9	8.64×10^{10}	2.62×10^{12}	3.15×10^{13}	3.15×10^{15}
$n \log_2(n)$	6.3×10^4	2.8×10^6	1.3×10^8	2.7×10^{10}	7.2×10^{10}	7.9×10^{11}	6.8×10^{13}
n^2	1000	7745.9	60000	293938	1618641	5612486	56124860
n^3	100	391.48	1532.61	4420	13803	31601	146589
2^n	19.93	25.85	31.75	36.33	41.24	44.84	51.48
$n!$	9	11	12	13	15	16	17

2. Supongamos que estamos comparando el desempeño de dos algoritmos de ordenamiento. Para entradas de tamaño n , el algoritmo A toma $8n^2$ operaciones mientras que el algoritmo B toma $64n \log_2(n)$. ¿Para qué valores de n es mejor el desempeño de A?
3. ¿Cuál es el valor más chico de n para el cual un algoritmo que toma $100n^2$ es más rápido que uno que toma 2^n (en la misma máquina)?
4. Demuestre que $2^n = O(n^2)$

2) A: $8n^2$, B: $64n \log_2(n)$

$$8n^2 < 64n \log_2(n)$$

$$\frac{8n^2}{8} < \frac{64n \log_2(n)}{8} \rightarrow n^2 < 8n \log_2(n) \rightarrow \text{Segun python } n \in (2, 43) \text{ para que el algoritmo A sea mejor}$$

3) $100n^2 < 2^n$

segun python a partir de 15

4) $2^n = O(n^2)$

$$2^n = Cn^2$$

$$C = \frac{2^n}{n^2}, n \neq 0$$