Lucía Lizardi Rodríguez

Dado que una computadora promedio tiene una velocidad de procesamiento específica (3.5GHz y 4GHz) Digamos 3.75GHz. Esos son 3.75 billones de operaciones por segundo y que generalmente los programas no trabajan a nivel de memoria, el valor dado anteriormente, se acota a 10^8 operaciones en un segundo … por lo tanto:

	1 segundo	1 min	1 hr	1 día	1 mes	1 año	1 siglo
$\log_2(n)$	2^{10^8}	$2^{6\times10^9}$	$2^{3.6\times10^{11}}$	$2^{8.64\times10^{12}}$	$2^{(8.64\times10^{12})(30)}$	$2^{8.64\times10^{12}(365)}$	$2^{(8.64\times10^{12})(365)(100)}$
$\sqrt{n}$	10^{16}	3.6×10^{9}	$(3.6\times10^{11})^2$	$(8.64\times10^{12})^2$	$((8.64\times10^{12})(30))^2$	$((8.64\times10^{12})(365))^2$	$(8.64\times10^{12})(365)(100)$
N	10^8	6×10^9	3.6×10^{11}	8.64×10^{12}	$(8.64\times10^{12})(30)$	$(8.64\times10^{12})\cdot(365)$	$(8.64\times10^{12})(365)(100)$
$n\log_2(n)$	$n\log_2(n)=10^8$	$n\log_2(n)=6\times10^9$	$n\log_2(n)=3.6\times10^{11}$	$n\log_2(n)=8.64\times10^{12}$	$n\log_2(n)=(8.64\times10^{12})(30)$	$n\log_2(n)=(8.64\times10^{12})(365)$	$n\log_2(n)=(8.64\times10^{12})(365)(100)$
n^2	$\sqrt{10^8}$	$\sqrt{6\times10^9}$	$\sqrt{3.6\times10^{11}}$	$\sqrt{8.64\times10^{12}}$	$\sqrt{(8.64\times10^{12})(30)}$	$\sqrt{(8.64\times10^{12})\cdot(365)}$	$\sqrt{(8.64\times10^{12})(365)(100)}$
n^3	$\sqrt[3]{10^8}$	$\sqrt[3]{6\times10^9}$	$\sqrt[3]{3.6\times10^{11}}$	$\sqrt[3]{8.64\times10^{12}}$	$\sqrt[3]{(8.64\times10^{12})(30)}$	$\sqrt[3]{8.64\times10^{12}\cdot365}$	$\sqrt[3]{(8.64\times10^{12})(365)(100)}$
2^n	$\log_2(10^8)$	$\log_2(6\times10^9)$	$\log_2(3.6\times10^{11})$	$\log_2(8.64\times10^{12})$	$\log_2(8.64\times10^{12}(30))$	$\log_2(8.64\times10^{12}\cdot365)$	$\log_2((8.64\times10^{12})(365)(100))$
$n!$	$n!=10^8$	$n!=6\times10^9$	$n!=3.6\times10^{11}$	$n!=8.64\times10^{12}$	$n!=(8.64\times10^{12})(30)$	$n!=(8.64\times10^{12})(365)$	$n!=(8.64\times10^{12})(365)(100)$

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? $1<n<43$

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)? $n>14$

4.

Demuestre que $2^n = O(n^2)$

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

$g(n) = n^2$
$f(n) = 2^n$

$0 \leq 2^n \leq cn^2$
$0 \leq cn^2 - 2^n$
$0 \leq \ln\left(\frac{cn^2}{2^n}\right)$

cualquer log es > 0

$\therefore$ c y n pueden tomar cualquer valor > 0