Introducción a la algoritmia II

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Maestría en Ciencias de la Computación
Análisis y Diseño de Algoritmos
MCOM 20300

Otoño 2020

Contenido

La importancia de la eficiencia

- Ya que las computadoras se hacen cada vez más y más rápidas, pudiera parecer que no vale la pena gastar nuestro tiempo tratando de diseñar algoritmos más eficientes.

La importancia de la eficiencia

- ¿No sería más fácil esperar a la siguiente generación de computadoras?

La importancia de la eficiencia

- Los principios establecidos anteriormente muestran que esto no es verdad.

Ejemplo de la importancia de la eficiencia I

Ejemplo 1

Suponga que para solucionar un problema particular tiene disponible un algoritmo exponencial y una computadora que puede correr este algoritmo sobre ejemplares de tamaño n en $10^{-4} \times 2^n$ segundos.

Ejemplo de la importancia de la eficiencia I

Ejemplo 1

Suponga que para solucionar un problema particular tiene disponible un algoritmo exponencial y una computadora que puede correr este algoritmo sobre ejemplares de tamaño n en $10^{-4} \times 2^n$ segundos.

n	$10^{-4} \times 2^n$ seg
10	$10^{-4} \times 2^{10}$ seg $=$ 0.1024 seg
20	$10^{-4} \times 2^{20}$ seg $=$ 104.86 seg
30	$10^{-4} \times 2^{30}$ seg $=$ 107,374.18 seg

Ejemplo de la importancia de la eficiencia II

Ejemplo 2

Suponiendo que se pudiera ejecutar una computadora sin interrupción y sin errores por una año completo. ¿Qué tamaño de ejemplar se puede resolver en un año?

Ejemplo de la importancia de la eficiencia II

Ejemplo 2

Suponiendo que se pudiera ejecutar una computadora sin interrupción y sin errores por una año completo. ¿Qué tamaño de ejemplar se puede resolver en un año?

Solución:

$$10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

Ejemplo de la importancia de la eficiencia II

Ejemplo 2

Suponiendo que se pudiera ejecutar una computadora sin interrupción y sin errores por una año completo. ¿Qué tamaño de ejemplar se puede resolver en un año?

Solución:

$$10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^4 \times 86400 \times 365$$

Ejemplo 2

Suponiendo que se pudiera ejecutar una computadora sin interrupción y sin errores por una año completo. ¿Qué tamaño de ejemplar se puede resolver en un año?

Solución:

$$10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^4 \times 86400 \times 365$$

$$n = \log_2(10^4 \times 86400 \times 365)$$

Ejemplo de la importancia de la eficiencia II

Ejemplo 2

Suponiendo que se pudiera ejecutar una computadora sin interrupción y sin errores por una año completo. ¿Qué tamaño de ejemplar se puede resolver en un año?

Solución:

$$10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^4 \times 86400 \times 365$$

$$n = \log_2(10^4 \times 86400 \times 365)$$

$$n = 38.19820$$

Ejemplo de la importancia de la eficiencia III

Ejemplo 3

Suponga que necesita resolver ejemplares más grandes que esto y que tiene el dinero suficiente para comprar una computadora nueva un ciento de veces más rápida que la primera. ¿Qué tamaño de ejemplar se puede resolver en un año con esta máquina 100 veces más rápida?

Ejemplo de la importancia de la eficiencia III

Ejemplo 3

Suponga que necesita resolver ejemplares más grandes que esto y que tiene el dinero suficiente para comprar una computadora nueva un ciento de veces más rápida que la primera. ¿Qué tamaño de ejemplar se puede resolver en un año con esta máquina 100 veces más rápida?

Solución:

$$\begin{aligned} 10^{-4} \times 2^n \text{ seg} &\text{ —— } t \text{ seg} \\ 10^2 t' &\text{ —— } t \text{ seg} \end{aligned}$$

Ejemplo de la importancia de la eficiencia III

Ejemplo 3

Suponga que necesita resolver ejemplares más grandes que esto y que tiene el dinero suficiente para comprar una computadora nueva un ciento de veces más rápida que la primera. ¿Qué tamaño de ejemplar se puede resolver en un año con esta máquina 100 veces más rápida?

Solución:

$$\begin{aligned} 10^{-4} \times 2^n \text{ seg} &\text{ —— } t \text{ seg} \\ 10^2 t' &\text{ —— } t \text{ seg} \end{aligned}$$

$$10^2 t' = \frac{t \times 10^{-4} \times 2^n \text{ seg}^2}{t \text{ seg}}$$

Ejemplo de la importancia de la eficiencia III

Ejemplo 3

Suponga que necesita resolver ejemplares más grandes que esto y que tiene el dinero suficiente para comprar una computadora nueva un ciento de veces más rápida que la primera. ¿Qué tamaño de ejemplar se puede resolver en un año con esta máquina 100 veces más rápida?

Solución:

$$\begin{aligned} 10^{-4} \times 2^n \text{ seg} &—— t \text{ seg} \\ 10^2 t' &—— t \text{ seg} \end{aligned}$$

$$10^2 t' = \frac{t \times 10^{-4} \times 2^n \text{ seg}^2}{t \text{ seg}}$$

$$10^2 t' = 10^{-4} \times 2^n \text{ seg}$$

Ejemplo de la importancia de la eficiencia III

Ejemplo 3

Suponga que necesita resolver ejemplares más grandes que esto y que tiene el dinero suficiente para comprar una computadora nueva un ciento de veces más rápida que la primera. ¿Qué tamaño de ejemplar se puede resolver en un año con esta máquina 100 veces más rápida?

Solución:

$$10^{-4} \times 2^n \text{ seg} \text{ —— } t \text{ seg}$$
$$10^2 t' \text{ —— } t \text{ seg}$$

$$10^2 t' = \frac{t \times 10^{-4} \times 2^n \text{ seg}^2}{t \text{ seg}}$$

$$10^2 t' = 10^{-4} \times 2^n \text{ seg}$$

$$t' = 10^{-2} \times 10^{-4} \times 2^n \text{ seg}$$

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

$$10^{-2} \times 10^{-4} \times 2^{n} \text{ seg} = 86400 \times 365 \text{ seg}$$

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

$$10^{-2} \times 10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^2 \times 10^4 \times 86400 \times 365$$

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

$$10^{-2} \times 10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^2 \times 10^4 \times 86400 \times 365$$

$$n = \log_2(10^2 \times \overbrace{10^4 \times 86400 \times 365}^{n_a})$$

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

$$10^{-2} \times 10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^2 \times 10^4 \times 86400 \times 365$$

$$n = \log_2(10^2 \times \overbrace{10^4 \times 86400 \times 365}^{n_a})$$

$$n = 44.84206$$

Ejemplo de la importancia de la eficiencia IV

Procedemos como al inicio del Ejemplo 2.

$$10^{-2} \times 10^{-4} \times 2^n \text{ seg} = 86400 \times 365 \text{ seg}$$

$$2^n = 10^2 \times 10^4 \times 86400 \times 365$$

$$n = \log_2(10^2 \times \overbrace{10^4 \times 86400 \times 365}^{n_a})$$

$$n = 44.84206$$

Con la nueva computadora a lo más resolveriamos ejemplares de tamaño

$$n = \log_2 100 + n_a < 7 + n_a$$

Ejemplo de la importancia de la eficiencia V

Ejemplo 4

- Suponga que mejor decide invertir en algoritmia y que habiendo gastado la misma cantidad de dinero, ha conseguido un algoritmo cúbico para solucionar el problema.

n	$10^{-2} \times n^3$ seg
10	$10^{-2} \times 10^3$ seg $=$ 10 seg
20	$10^{-2} \times 20^3$ seg $=$ 80 seg
30	$10^{-2} \times 30^3$ seg $=$ 270 seg

Ejemplo de la importancia de la eficiencia V

Ejemplo 4

- Imagine por ejemplo, que usando la primera máquina con el nuevo algoritmo se pude resolver un ejemplar de tamaño n en $10^{-2} \times n^3$ segundos.

n	$10^{-2} \times n^3$ seg
10	$10^{-2} \times 10^3$ seg $=$ 10 seg
20	$10^{-2} \times 20^3$ seg $=$ 80 seg
30	$10^{-2} \times 30^3$ seg $=$ 270 seg

Ejemplo de la importancia de la eficiencia VI

Ejemplo 5

¿Qué tamaño de ejemplar se puede resolver con el algoritmo cúbico del Ejemplo 4 en un día?

Solución:

$$10^{-2} \times n^3 \text{ seg} = 86400 \text{ seg}$$

$$n^3 = 10^2 \times 86400$$

$$n = (10^2 \times 86400)^{\frac{1}{3}}$$

$$n = 205.1971$$

Ejercicios sobre la importancia de la eficiencia I

1. ¿Qué tamaño de ejemplar se puede resolver con el algoritmo cúbico del Ejemplo 4 en un año?
2. Demuestre que si puede hacer uso del nuevo algoritmo y de la máquina 100 veces más rápida, entonces podrá resolver ejemplares cuatro o cinco veces más grandes que con sólo el nuevo algoritmo, en la misma cantidad de tiempo, dé el factor exacto.
3. Suponga que mide el desempeño de un programa, quizas usando alguna clase de traza en tiempo de ejecución, luego optimiza las partes muy usadas del código. No obstante se asegura de no cambiar el algoritmo subyacente. ¿Qué esperaría obtener: (a) una ganancia en eficiencia mediante un factor constante, o (b) una ganancia en eficiencia que es proporcionalmente mayor conforme el tamaño del ejemplar aumenta? Justifique su respuesta.

Ejercicios sobre la importancia de la eficiencia II

4. Un algoritmo de ordenamiento consume 1 segundo para ordenar 1000 artículos en su computadora local.¿Cuánto consumiría para ordenar 10000 artículos si (a) cree que el algoritmo consume tiempo aproximadamente proporcional a n^2 y (b) cree que el algoritmo consume tiempo aproximadamente proporcional a $n \log n$?
5. Dos algoritmos consumen respectivamente n^2 días y 2^n segundos para resolver un ejemplar de tamaño n. ¿Cuál es el tamaño del ejemplar más pequeño para el cual el primer algoritmo supera al segundo? ¿Aproximadamente cuánto tiempo consumiría dicho ejemplar en ser resuelto?

Un algoritmo lineal para ordenamiento

- Se puede demostrar que ningún algoritmo de ordenamiento, que trabaje comparando los elementos a ser ordenados puede ser más rápido que el orden de $n \log n$.

Un algoritmo lineal para ordenamiento

- No obstante, se pueden encontrar otros algoritmos de ordenamiento más eficientes para casos **muy especiales**.

Un algoritmo lineal para ordenamiento

- Suponga por ejemplo que los elementos a ordenar son enteros que se sabe están en el rango de 1 a 10000. Entonces el siguiente algoritmo puede usarse.

Un algoritmo lineal para ordenamiento

procedure $\textsc{Casillas}(T[1 \ldots n])$

array $U[1 \ldots 10000]$
for $k \leftarrow 1$ **to** 10000
do $U[k] \leftarrow 0$
for $i \leftarrow 1$ **to** n
do $k \leftarrow T[i]$
    $U[k] \leftarrow U[k] + 1$
$i \leftarrow 0$
for $k \leftarrow 1$ **to** 10000
do while $U[k] \neq 0$
    do $i \leftarrow i + 1$
        $T[i] \leftarrow k$
        $U[k] \leftarrow U[k] - 1$

Ejercicios sobre el algoritmo Casillas

6. Analice el funcionamiento del algoritmo.
7. Explique su funcionamiento.
8. Impleméntelo en python.
9. Para probarlo genere aleatoriamente números entre 1 y 10000.
10. Muestre que ordenamiento por CASILLAS consume un tiempo en el orden de n para ordenar n elementos que están dentro de las cotas.