Análisis y diseño de algoritmos
Apoyo a las matemáticas

Juan Morales, Victor M. Sánchez y José Luis Verdú

12 de enero de 2025

1. Series Aritméticas

Definición 1.1. *Una serie de números* $a_0, a_1, \ldots, a_n$ *es aritmética si la diferencia entre dos términos sucesivos cualesquiera es constante. Esta constante se suele denominar* diferencia *o* paso*.*

Ejemplo 1.1. $1, 2, 3, \ldots, 10$ *es una serie arimética de paso 1.*

Ejemplo 1.2. $10, 8, 6, \ldots, -8$ *es una serie arimética de paso -2.*

Ejemplo 1.3. $1, 1, 1, \ldots, 1$ *es una serie arimética de paso 0.*

Dado que el paso (p) de una serie aritmética es constante el término k-ésimo puede escribirse como:

$$a_k = a_0 + kp$$

Por tanto, toda serie aritmética puede escribirse como:

$$a_0, a_0 + p, a_0 + 2p, \ldots, a_0 + np$$

1.1. Suma de series aritméticas

Sea

$$S = \sum_{i=0}^{n} a_i$$

la suma de una serie aritmética.

$$S = \quad a_0 \quad + \quad (a_0 + p) \quad + \ldots + (a_0 + (n-1)p) + \; (a_0 + np)$$

Si lo volvemos escribir pero empezando por el final

$$S = \; (a_0 + np) \; + (a_0 + (n-1)p) + \ldots + \quad (a_0 + p) \quad + \quad a_0$$

sumando las dos ecuaciones anteriores

$$2S = (2a_0 + np) + \quad (2a_0 + np) \quad + \ldots + \quad (2a_0 + np) \quad + (2a_0 + np)$$

y como hay $(n+1)$ términos, la suma queda

$$2S = (n+1)(2a_0 + np)$$

Despejando S,

$$S = \sum_{i=0}^{n} a_i = (n+1)(a_0 + \frac{np}{2})$$

Y ahora, recordando que $a_n = a_0 + np$, la ecuación puede escribirse como:

$$\sum_{i=0}^{n} a_i = (n+1)(a_0 + \frac{a_n - a_0}{2}) = \frac{(n+1)}{2}(a_0 + a_n)$$

1.1.1. Ejercicios

$$\sum_{i=0}^{n} i = \frac{(n+1)}{2} n$$

$$\sum_{i=0}^{n} (n-i) = \frac{(n+1)}{2} n$$

$$\sum_{i=0}^{n/2} (n-2i) = \frac{\frac{n}{2}+1}{2}(n+0) = \frac{n+2}{4} n$$

2. Series geométricas

Definición 2.1. *Una serie de números* $a_0, a_1, \ldots, a_n$ *es geométrica si cada número (excepto el primero) se obtiene multiplicando el anterior por una constante. Esta constante se suele denominar* razón.

Ejemplo 2.1. $1, 2, 4, 8, 16, \ldots$ *es una serie geométrica de razón* 2

Ejemplo 2.2. $10, 5, 2.5, 1.25, \ldots$ *es una serie geométrica de razón* $1/2$

Ejemplo 2.3. $1, 1, 1, 1, \ldots$ *es una seria geométrica de razón* 1

Ejemplo 2.4. $1, -1, 1, -1, \ldots$ *es una serie geométrica de razón* -1

Dado que la razón (r) de una serie geométrica es constante, el término k-ésimo puede escribirse como:

$$a_k = a_0 r^k$$

Con lo que toda serie geométrica puede escribirse como:

$$a_0 r^0, a_0 r^1, a_0 r^2, \ldots$$

2.1. Suma de series geométricas

Sea

$$S = \sum_{i=0}^{n} a_i$$

La suma de una serie geométrica.

$$S = a_0 + a_0 r + \ \dots \ + a_0 r^{n-1} + a_0 r^n$$

Si multiplicamos la ecución por la razón (r) tenemos:

$$rS = \qquad a_0 r + a_0 r^2 + \quad \dots \quad + a_0 r^n + a_0 r^{n+1}$$

y ahora restamos las dos ecuaciones anteriores, nos queda

$$S - rS = a_0 + \ 0 \ + \ 0 \ + \ \dots \ + \ 0 \ - a_0 r^{n+1}$$

Y despejando S nos queda la ecuación:

$$S = \sum_{i=0}^{n} a_i = a_0 \frac{1 - r^{n+1}}{1 - r}$$

¡Atención! la fórmula anterior solo sirve si $r \neq 1$, si $r = 1$.

$$\sum_{i=0}^{n} a_i = \sum_{i=0}^{n} a_0 1^i = a_0 \sum_{i=0}^{n} 1 = a_0(n+1)$$

2.1.1. Ejercicio

$$\sum_{i=0}^{n} 2^i = \frac{1 - 2^{n+1}}{1 - 2} = 2^{n+1} - 1$$
$$\sum_{i=0}^{n} \frac{1}{2^i} = \frac{1 - \frac{1}{2^{n+1}}}{1 - \frac{1}{2}} = 2 - \frac{1}{2^n}$$
$$\sum_{i=0}^{\infty} \frac{1}{2^i} = \lim_{n \to \infty} \left(2 - \frac{1}{2^n} \right) = 2$$

3. Series aritmetico-geométricas

Definición 3.1. *Una serie aritmetico-geometrica es una serie resultado del producto, témino a término, de una serie aritmetica y una serie geométrica.*

Ejemplo 3.1. $\frac{0}{1}, \frac{1}{2}, \frac{2}{4}, \frac{3}{8}, \frac{4}{16} \dots$ *es una serie aritmetico-geométrica de paso* 1 *y razón* 1/2*,*

3.1. Suma de series aritmetico-geometricas

Existe una fórmula general, pero es demasiado complicada para un uso ocasional. Lo más adecuado es aplicar la técnica para cada caso concreto que queramos sumar.

3.1.1. Ejercicios

Ejercicio 3.1. *Sumar la serie*

$$S = \sum_{i=0}^{n} \frac{i}{2^i}$$

Solución. Esta es una serie aritmético-geometrice de razón 1/2 y paso 1.

Para hacer la suma se procede de forma similar a como se sumaban las series geométricas

$$S = \tfrac{0}{2^0} + \tfrac{1}{2^1} + \tfrac{2}{2^2} + \dots + \tfrac{n-1}{2^{n-1}} + \tfrac{n}{2^n}$$

primero multiplicamos la ecuación por la razón (1/2)

$$\frac{1}{2}S = \tfrac{0}{2^1} + \tfrac{1}{2^2} + \tfrac{1}{2^3} + \dots + \tfrac{n-1}{2^n} + \tfrac{n}{2^{n+1}}$$

Y, ahora, a la primera ecuación le restamos la segunda

$$S - \frac{1}{2}S = 0 + \tfrac{1}{2} + \tfrac{1}{2^2} + \dots + \tfrac{1}{2^n} - \tfrac{n}{2^{n+1}}$$

Operando un poco vemos que aparece la suma de una serie geométrica

$$\frac{1}{2}S = \frac{1}{2}\left[1 + \frac{1}{2} + \dots + \frac{1}{2^{n-1}}\right] - \frac{n}{2^{n+1}}$$

Hacemos la suma de la serie geométrica

$$\frac{1}{2}S = \frac{1}{2}\left[2 - \frac{1}{2^n}\right] - \frac{n}{2^{n+1}}$$

y, operando un poco más, obtenemos el resultado

$$S = 2 - \frac{1}{2^{n-1}} - \frac{n}{2^n} = 2 - \frac{2+n}{2^n}$$

Ejercicio 3.2. *Sumar la serie*

$$S = \sum_{i=0}^{\infty} \frac{i}{2^i}$$

Solución.

$$\sum_{i=0}^{\infty} \frac{i}{2^i} = \lim_{n\to\infty} \sum_{i=0}^{n} \frac{i}{2^i}$$
$$= \lim_{n\to\infty} \left(2 - \frac{2+n}{2^n}\right) = 2$$

Ejercicio 3.3. *Sumar la serie*

$$S = \sum_{i=0}^{n} (n-i)2^i$$

Solución. Se procede de forma similar a como se sumaban las serie geométricas

$$S = (n-0)2^0 + (n-1)2^1 + (n-2)2^2 + \quad \dots \quad + \quad 1\,2^{n-1} \quad + \quad 0\,2^n$$

Multiplicamos por la razón, 2 en este caso

$$2S = \qquad (n-0)2^1 + (n-1)2^2 + (n-2)2^3 + \quad \dots \quad + \quad 1\,2^n \quad + 0\,2^{n+1}$$

A la segunda ecuación le restamos la primera

$$2S - S = \quad -n \quad + \quad 2^1 \quad + \quad 2^2 \quad + \quad 2^3 \quad + \quad \dots \quad + \quad 2^n \quad + \quad 0$$

Operando vemos que sale una suma de una serie geométrica que ya conocemos

$$S = -n + 2\left[2^0 + 2^1 + 2^2 + \dots + 2^{n-1}\right] = 2^{n+1} - 2 - n$$

4. Acotando series

Ejercicio 4.1. *demostrar que*

$$\sum_{i=1}^{n} \frac{1}{2^i} \in \Theta(1) \tag{1}$$

Solución. Por una parte:

$$\sum_{i=1}^{n} \frac{1}{2^i} \geq \frac{1}{2} \in \Omega(1) \tag{2}$$

y por otra parte:

$$\sum_{i=1}^{n} \frac{1}{2^i} < \sum_{i=1}^{\infty} \frac{1}{2^i} = 1 \in O(1) \tag{3}$$

Por lo que

$$\sum_{i=1}^{n} \frac{1}{2^i} \in \Theta(1) \tag{4}$$

Ejercicio 4.2. *demostrar que*

$$\sum_{i=0}^{n} i^c \in \Theta(n^{c+1}) \qquad c \geq 1$$

Solución. Por una parte tenemos que

$$\sum_{i=0}^{n} i^c \leq \sum_{i=0}^{n} n^c = n^c \sum_{i=0}^{n} 1 \in O(n^{c+1})$$

Por otra parte, tomando $m = 2\lfloor n/2 \rfloor$(el par justo inferior o igual a n),

$$\sum_{i=0}^{n} i^c \geq \sum_{i=0}^{m} i^c \geq \sum_{i=m/2}^{m} i^c \geq \sum_{i=m/2}^{m} \left(\frac{m}{2}\right)^c = \left(\frac{m}{2}\right)^c \sum_{i=m/2}^{m} 1 = \frac{m^c}{2^c}\left(\frac{m}{2}+1\right) \in \Omega(n^{c+1})$$

Por tanto, como

$$\sum_{i=0}^{n} i^c \in O(n^{c+1}) \quad \wedge \quad \sum_{i=0}^{n} i^c \in \Omega(n^{c+1}) \Rightarrow \sum_{i=0}^{n} i^c \in \Theta(n^{c+1})$$

Ejercicio 4.3. *demostrar que*

$$\sum_{i=1}^{n} \log(i) \in \Theta(n \log(n))$$

Solución. Por una parte tenemos que

$$\sum_{i=1}^{n} \log(i) \leq \sum_{i=1}^{n} \log(n) = n \log(n) \in O(n \log(n))$$

Por otra parte, tomando $m = \lfloor \log(n) \rfloor$ (nótese que $m \leq \log(n) \leq m+1$)

$$\sum_{i=1}^{n} \log(i) \geq \sum_{i=2^{m-1}}^{2^m-1} \log(i) \geq \sum_{i=2^{m-1}}^{2^m-1} \log(2^{m-1}) = (m-1) \sum_{i=2^{m-1}}^{2^m-1} 1$$

$$= (m-1)\, 2^{m-1} \geq (\log(n)-2)\, 2^{\log(n)-2} = (\log(n)-2)\, n\, \frac{1}{4} \in \Omega(n \log(n))$$

Por tanto, como

$$\sum_{i=1}^{n} \log(i) \in O(n \log(n)) \quad \wedge \quad \sum_{i=0}^{n} \log(i) \in \Omega(n \log(n)) \Rightarrow \sum_{i=1}^{n} \log(i) \in \Theta(n \log(n))$$

Ejercicio 4.4. *demostrar que*

$$\sum_{i=0}^{\log_a(n)} b^i \in \Theta(n^{\log_a(b)})$$

Solución.

$$\sum_{i=0}^{\log_a(n)} b^i = \frac{b^{\log_a(n)+1} - 1}{b-1}$$
$$= \frac{bb^{\log_a(n)} - 1}{b-1}$$
$$= \frac{bn^{\log_a(b)} - 1}{b-1} = \Theta(n^{\log_a(b)})$$

Ejercicio 4.5. *demostrar que*

$$\sum_{i=0}^{m} a_i n^i \in \Theta(n^m), a_i \geq 0, a_m > 0$$

Solución. Sea $a = \text{máx}\{a_i\}$.

Por una parte tenemos que

$$\sum_{i=0}^{m} a_i n^i \leq \sum_{i=0}^{m} a n^i = a \sum_{i=0}^{m} n^i \leq a \sum_{i=0}^{m} n^m = a n^m (m+1) \in O(n^m)$$

Por otra parte

$$\sum_{i=0}^{m} a_i n^i \geq a_m n^m \in \Omega(n^m)$$

5. Calculando relaciones de recurrencia

Ejercicio 5.1. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + 2T(n/2) & n > 1 \end{cases}$$

Solución.

$$T(n) \stackrel{1}{=} 1 + 2T(\frac{n}{2})$$
$$\stackrel{2}{=} 1 + 2 + 2^2 T(\frac{n}{2^2})$$
$$\stackrel{3}{=} 1 + 2 + 2^2 + 2^3 T(\frac{n}{2^3})$$
$$\dots$$
$$\stackrel{k}{=} \sum_{i=0}^{k-1} 2^i + 2^k T(\frac{n}{2^k})$$

Paramos cuando $\frac{n}{2^k} = 1$, o sea cuando $k = \lfloor \log_2 n \rfloor$. Por lo que

$$T(n) = \sum_{i=0}^{\lfloor \log_2 n \rfloor - 1} 2^i + 2^{\lfloor \log_2 n \rfloor} = \sum_{i=0}^{\lfloor \log_2 n \rfloor} 2^i = 2^{\lfloor \log_2 n \rfloor + 1} - 1$$

Es fácil ver que (lo usaremos en otros ejercicios)

$$\lfloor f(x) \rfloor \in \Theta(f(x))$$

Ya que

$$\begin{aligned} \lfloor f(x) \rfloor &\leq f(x) \in O(f(x)) \\ \lfloor f(x) \rfloor &\geq f(x) - 1 \in \Omega(f(x)) \end{aligned}$$

En particular,

$$\lfloor \log_2 n \rfloor \in \Theta(\log_2 n)$$

Por tanto,

$$T(n) = 2^{\lfloor \log_2 n \rfloor + 1} - 1 \in \Theta(2^{\log_2 n + 1} - 1) = \Theta(2n - 1) = \Theta(n)$$

Ejercicio 5.2. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ n + 7T(n/7) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned} T(n) &\overset{1}{=} n + 7T(\frac{n}{7}) \\ &\overset{2}{=} n + 7\frac{n}{7} + 7^2 T(\frac{n}{7^2}) \\ &\overset{3}{=} n + 7\frac{n}{7} + 7^2 \frac{n}{7^2} + 7^3 T(\frac{n}{7^3}) \\ &\dots \\ &\overset{k}{=} kn + 7^k T(\frac{n}{7^k}) \end{aligned}$$

Paramos cuando $\frac{n}{7^k} = 1$, o sea cuando $k = \lfloor \log_7 n \rfloor$

$$T(n) = n\lfloor \log_7 n \rfloor + 7^{\lfloor \log_7 n \rfloor} \in \Theta(n \log_7 n + n) = \Theta(n \log n)$$

Ejercicio 5.3. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ n^2 + 9T(n/3) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned}
T(n) &\overset{1}{=} n^2 + 9T(\frac{n}{3}) \\
&\overset{2}{=} n^2 + 9(\frac{n}{3})^2 + 9^2T(\frac{n}{3^2}) \\
&\overset{3}{=} n^2 + 9(\frac{n}{3})^2 + 9^2(\frac{n}{3^2})^2 + 9^3T(\frac{n}{3^3}) \\
&\dots \\
&\overset{k}{=} kn^2 + 9^kT(\frac{n}{3^k})
\end{aligned}$$

Paramos cuando $\frac{n}{3^k} = 1$, o sea cuando $k = \lfloor \log_3 n \rfloor$

$$T(n) = n^2\lfloor \log_3 n \rfloor + 9^{\lfloor \log_3 n \rfloor} \in \Theta(n^2 \log_3 n + n^2) = \Theta(n^2 \log n)$$

Ejercicio 5.4. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ n^3 + 8T(n/2) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned}
T(n) &\overset{1}{=} n^3 + 8T(\frac{n}{2}) \\
&\overset{2}{=} n^3 + 8(\frac{n}{2})^3 + 8^2T(\frac{n}{2^2}) \\
&\overset{3}{=} n^3 + 8(\frac{n}{2})^3 + 8^2(\frac{n}{2^2})^3 + 8^3T(\frac{n}{2^3}) \\
&\dots \\
&\overset{k}{=} n^3k + 8^kT(\frac{n}{2^k})
\end{aligned}$$

Paramos cuando $\frac{n}{2^k} = 1$, o sea cuando $\lfloor k = \log n \rfloor$

$$T(n) = n^3\lfloor \log n \rfloor + 8^{\lfloor \log n \rfloor} \in \Theta(n^3 \log n + n^3) \in \Theta(n^3 \log n)$$

Ejercicio 5.5. *(difícil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ n^{3/2}\log n + 49T(n/25) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned}
T(n) &\overset{1}{=} n^{\frac{3}{2}}\log n + 49T(\frac{n}{25})\\
&\overset{2}{=} n^{\frac{3}{2}}\log n + 49(\frac{n}{25})^{\frac{3}{2}}\log\frac{n}{25} + 49^2T(\frac{n}{25^2})\\
&\overset{3}{=} n^{\frac{3}{2}}\log n + 49(\frac{n}{25})^{\frac{3}{2}}\log\frac{n}{25} + 49^2(\frac{n}{25^2})^{\frac{3}{2}}\log\frac{n}{25^2} + 49^3T(\frac{n}{25^3})\\
&\dots\\
&\overset{k}{=} \sum_{i=0}^{k-1} 49^i(\frac{n}{25^i})^{\frac{3}{2}}\log\frac{n}{25^i} + 49^kT(\frac{n}{25^k})\\
&\overset{k}{=} n^{\frac{3}{2}}\sum_{i=0}^{k-1}\frac{49^i}{125^i}\log\frac{n}{25^i} + 49^kT(\frac{n}{25^k})\\
&\overset{k}{=} n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\frac{49^i}{125^i} - n^{\frac{3}{2}}\sum_{i=0}^{k-1}\frac{49^i}{125^i}\log 25^i + 49^kT(\frac{n}{25^k})\\
&\overset{k}{=} n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\left(\frac{49}{125}\right)^i - n^{\frac{3}{2}}\log 25\sum_{i=0}^{k-1} i\left(\frac{49}{125}\right)^i + 49^kT(\frac{n}{25^k})
\end{aligned}$$

Paramos cuando $\frac{n}{25^k} = 1$, o sea cuando $k = \lfloor n \rfloor$

Vamos a simplificar un poco antes de sustituir.

Como $\frac{49}{125} < 1$ sabemos que las series $\sum_{i=0}^{\infty}\left(\frac{49}{125}\right)^i$ y $\sum_{i=0}^{\infty} i\left(\frac{49}{125}\right)^i$ convergen a un número positivo. Llamemos, respectivamente, K_1 y K_2 a esos números.

Por tanto, por una parte,

$$\begin{aligned}
&n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\left(\frac{49}{125}\right)^i - n^{\frac{3}{2}}\log 25\sum_{i=0}^{k-1} i\left(\frac{49}{125}\right)^i\\
&\le n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\left(\frac{49}{125}\right)^i\\
&\le K_1 n^{\frac{3}{2}}\log n \in O(n^{\frac{3}{2}}\log n)
\end{aligned}$$

Y, por otra parte

$$\begin{aligned}
&n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\left(\frac{49}{125}\right)^i - n^{\frac{3}{2}}\log 25\sum_{i=0}^{k-1} i\left(\frac{49}{125}\right)^i\\
&\ge n^{\frac{3}{2}}\log n - K_2 n^{\frac{3}{2}}\log 25 \in \Omega(n^{\frac{3}{2}}\log n)
\end{aligned}$$

Por tanto,

$$n^{\frac{3}{2}}\log n\sum_{i=0}^{k-1}\left(\frac{49}{125}\right)^i - n^{\frac{3}{2}}\log 25\sum_{i=0}^{k-1} i\left(\frac{49}{125}\right)^i \in \Theta(n^{\frac{3}{2}}\log n)$$

Así que,

$$\begin{aligned} T(n) &\in \Theta(n^{\frac{3}{2}}\log n + 49^{\lfloor \log_{25} n\rfloor}) \\ &= \Theta(n^{\frac{3}{2}}\log n + n^{\log_{25}(49)}) \\ &= \Theta(n^{\frac{3}{2}}\log n) \end{aligned}$$

Ya que $\frac{3}{2} > \log_{25} 49 \approx 1.2090\ldots$

Ejercicio 5.6. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + T(n-1) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned} T(n) &\overset{1}{=} 1 + T(n-1) \\ &\overset{2}{=} 2 + T(n-2) \\ &\overset{3}{=} 3 + T(n-3) \\ &\ldots \\ &\overset{k}{=} k + T(n-k) \end{aligned}$$

Paramos cuando $n-k=1$, o sea cuando $k = n-1$

$$T(n) = n - 1 + 1 \in \Theta(n)$$

Ejercicio 5.7. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + 2T(n-1) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned} T(n) &\overset{1}{=} 1 + 2T(n-1) \\ &\overset{2}{=} 1 + 2(1 + 2T(n-2)) = 1 + 2 + 2^2T(n-2) \\ &\overset{3}{=} 1 + 2 + 2^2(1 + 2T(n-3)) = 1 + 2 + 2^2 + 2^3T(n-3) \\ &\ldots \\ &\overset{k}{=} \sum_{i=0}^{k-1} 2^i + 2^kT(n-k) \end{aligned}$$

Paramos cuando $n - k = 1$, o sea cuando $k = n - 1$

$$T(n) = \sum_{i=0}^{n-2} 2^i + 2^{n-1} = (2^{n-1} - 1) + 2^{n-1} \in \Theta(2^n)$$

Ejercicio 5.8. *(fácil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ n^c + T(n-1) & n > 1 \end{cases}$$

Solución.

$$\begin{aligned} T(n) &\stackrel{1}{=} n^c + T(n-1) \\ &\stackrel{2}{=} n^c + (n-1)^c + T(n-2) \\ &\stackrel{3}{=} n^c + (n-1)^c + (n-2)^c + T(n-3) \\ &\dots \\ &\stackrel{k}{=} \sum_{i=0}^{k-1} (n-i)^c + T(n-k) \end{aligned}$$

Paramos cuando $n - k = 1$, o sea cuando $k = n - 1$

$$T(n) = \sum_{i=0}^{n-2} (n-i)^c + 1 = \sum_{j=2}^{n} j^c + 1 \in \Theta(n^{1+c})$$

Ejercicio 5.9. *(difícil) Calcular:*

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + T(\sqrt{n}) & n > 1 \end{cases}$$

Solución. Hay que tener en cuenta que en estas relaciones de recurrencia estamos trabajando con números enteros, por lo que la raíz cuadrada que aparece es entera (redondea a la baja). O sea, que deberíamos haberla escrito como:

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + T(\sqrt{n}) & n > 1 \end{cases}$$

Para facilitarnos el trabajo vamos a definir la siguiente relación de recurrencia sobre números reales:

$$T'(n) = \begin{cases} 1 & n < 2 \\ 1 + T'(\sqrt{n}) & n \geq 2 \end{cases}$$

Es fácil demostrar por inducción que $T(n) = T'(n)$.

Vamos a ver que si n y $n' \in [2^{2^k}, 2^{2^{(k+1)}}[$ entonces $T(n) = T'(n') = k + 2$. Lo vamos a hacer por inducción en k.

Si $k = 0$ es muy fácil comprobar que si n y $n' \in [2, 2^2[$ entonces $T(n) = T'(n') = 2$. De paso también es fácil comprobar que $T(1) = T'(1) = 1$.

Supongamos que la condición es cierta para un $k > 0$, vamos a ver que pasa para $k + 1$. O sea, que pasa cuando n y $n' \in [2^{2^{k+1}}, 2^{2^{(k+1)}}[$.

Es evidente que como n y $n' \in [2^{2^{(k+1)}}, 2^{2^{(k+2)}}[$ entonces $\sqrt{n}$ y $\sqrt{n'} \in [2^{2^k}, 2^{2^{(k+1)}}[$. Además, como $2^{2^{(k+1)}}$ es un número entero, $\lfloor\sqrt{n}\rfloor \in [2^{2^k}, 2^{2^{(k+1)}}[$. Entonces, aplicando la hipótesis de inducción tenemos que $T(\lfloor\sqrt{n}\rfloor) = T'(\sqrt{n}) = k + 2$ y, por tanto, $T(n) = T'(n') = k + 3$.

A partir de ahora vamos a trabajar con T'.

Desplegando:

$$\begin{aligned}
T'(n) &\overset{1}{=} 1 + T'(n^{\frac{1}{2}}) \\
&\overset{2}{=} 2 + T'(n^{\frac{1}{2^2}}) \\
&\overset{3}{=} 3 + T'(n^{\frac{1}{2^3}}) \\
&\dots \\
&\overset{k}{=} k + T'(n^{\frac{1}{2^k}})
\end{aligned}$$

Paramos cuando $n^{\frac{1}{2^k}} < 2$ o sea cuando,

$$\begin{aligned}
\frac{1}{2^k}\log n &< \log 2 = 1 \\
2^k &> \log n \\
k &> \log(\log n)
\end{aligned}$$

Así que k será el menor entero que cumpla esa relación. O sea, $k = \lceil\log\log n\rceil$.

Entonces,

$$T(n) = T'(n) = \lceil\log\log n\rceil + 1 \in \Theta(\log\log n + 2) = \Theta(\log\log n)$$

Ejercicio 5.10. *(difícil) Resolver*

$$T(n) = \begin{cases} 1 & n = 1 \\ n^c + hT(n/a) & n > 1 \end{cases}$$

Solución. Desplegando:

$$
\begin{aligned}
T(n) &\overset{1}{=} n^c + hT\left(\frac{n}{a}\right) \\
&\overset{2}{=} n^c + h\left(\frac{n}{a}\right)^c + h^2T\left(\frac{n}{a^2}\right) \\
&\overset{3}{=} n^c + h\left(\frac{n}{a}\right)^c + h^2\left(\frac{n}{a^2}\right)^c + h^3T\left(\frac{n}{a^3}\right) \\
&\dots \\
&\overset{k}{=} n^c \sum_{i=0}^{k-1}\left(\frac{h}{a^c}\right)^i + h^kT\left(\frac{n}{a^k}\right)
\end{aligned}
$$

Pararemos cuando $\frac{n}{a^k} = 1$, o sea cuando $k = \lfloor \log_a n \rfloor$

$$
\overset{k}{=} n^c \sum_{i=0}^{\lfloor \log_a n \rfloor - 1}\left(\frac{h}{a^c}\right)^i + h^{\lfloor \log \rfloor_a n}
$$

El sumatorio del primer término es una serie geométrica para el caso $h \neq a^c$ y una serie constante si $h = a^c$, por lo tanto, para resolverlo distinguimos estos tres casos posibles:

- $h < a^c$:
  Se trata de una serie geométrica de *razón* menor a la unidad. Esta serie, si la extendemos al infinito, converge. Llamaremos K a ese valor.

  $$1 \leq \sum_{i=0}^{\lfloor \log_a n \rfloor - 1}\left(\frac{h}{a^c}\right)^i < \sum_{i=0}^{\infty}\left(\frac{h}{a^c}\right)^i = K$$

  Por lo que

  $$\sum_{i=0}^{\lfloor \log_a n \rfloor - 1}\left(\frac{h}{a^c}\right)^i \in \Theta(1)$$

  Por tanto,

  $$T(n) \in \Theta(n^c + h^{\lfloor \log_a n \rfloor}) = \Theta(n^c + h^{\log_a n}) = \Theta(n^c + n^{\log_a h})$$

  Y como $c > \log_a(h)$ (ya que estamos suponiendo que $h < a^c$)

  $$T(n) \in \Theta(n^c)$$

- $h = a^c$:
  En este caso $\frac{h}{a^c} = 1$, por lo que

  $$\sum_{i=0}^{\lfloor \log_a n \rfloor - 1}\left(\frac{h}{a^c}\right)^i = \sum_{i=0}^{\lfloor \log_a n \rfloor - 1} 1^i = \lfloor \log_a n \rfloor$$

Por tanto,

$$T(n) = n^c \lfloor \log_a n \rfloor + h^{\lfloor \log_a n \rfloor} \in \Theta(n^c \log_a n + n^{\log_a h})$$

Y como $c = \log_a(h)$ (ya que estamos suponiendo que $h = a^c$)

$$T(n) \leq n^c \log_a n + n^c \in O(n^c \log_a n).$$

- $h > a^c$:
  Se trata de una serie geométrica de *razón* mayor a la unidad.

$$\sum_{i=0}^{\lfloor \log_a n \rfloor - 1} \left(\frac{h}{a^c}\right)^i = \frac{(\frac{h}{a^c})^{\lfloor \log_a n \rfloor} - 1}{\frac{h}{a^c} - 1} \in \Theta\left(\left(\frac{h}{a^c}\right)^{\lfloor \log_a n \rfloor}\right)$$

  Por tanto,

$$\begin{aligned} T(n) &\in \Theta\left(n^c \left(\frac{h}{a^c}\right)^{\lfloor \log_a n \rfloor} + h^{\lfloor \log_a n \rfloor}\right) \\ &= \Theta\left(n^c \left(\frac{h}{a^c}\right)^{\log_a n} + h^{\log_a n}\right) \\ &= \Theta\left(n^c \frac{n^{\log_a h}}{(a^{\log_a n})^c} + n^{\log_a h}\right) \\ &= \Theta\left(n^{\log_a h} + n^{\log_a h}\right) \\ &= \Theta(n^{\log_a h}) \end{aligned}$$

Ejercicio 5.11. *(difícil) Resolver*

$$T(n) = \begin{cases} 1 & n = 1 \\ \log_a n + aT(n/a) & n > 1 \end{cases} \qquad a \in \mathbb{N}^{>1}$$

Solución. Aplicando sustituciones sucesivas se llega a la siguiente expresión general que depende del número k de sustituciones que se pueden realizar (su veracidad se puede demostrar fácilmente por *inducción matemática* sobre k):

$$T(n) = a^k T\left(\frac{n}{a^k}\right) + \sum_{j=0}^{k-1} a^j \log_a \frac{n}{a^j} \qquad \forall k \in \mathbb{N}; 1 \leq k \leq \lfloor \log_a n \rfloor$$

En virtud de las propiedades de los logaritmos y operando convenientemente, podemos desdoblar el sumatorio en dos más sencillos:

$$T(n) = a^k T\left(\frac{n}{a^k}\right) + \log_a(n) \sum_{j=0}^{k-1} a^j - \sum_{j=0}^{k-1} j a^j \tag{5}$$

Para que la recursión termine interesa sustituir k por $\lfloor \log_a n \rfloor$, pero antes de hacerlo, simplificaremos la expresión centrándonos en ambos sumatorios:

- El primer sumatorio es una serie geométrica de *razón* $a > 1$; puesto que su primer elemento es 1, tenemos:

$$\sum_{j=0}^{k-1} a^j = \frac{1 - a^k}{1 - a} \tag{6}$$

- El segundo sumatorio es una serie aritmético-geométrica de *razón* $a > 1$ y *diferencia* 1. Para sumarla procedemos según se ha explicado en la sección 3.1

$$S = \sum_{j=0}^{k-1} j a^j$$

expandiendo este sumatorio se tiene:

$$S = \quad 0\,a^0 \quad + \quad 1\,a^1 \quad + \quad 2\,a^2 \quad + \quad \ldots \quad + (k-1)\,a^{k-1}$$

multiplicando ambos lados del la igualdad por la razón:

$$a\,S = \qquad\qquad 0\,a^1 \quad + \quad 1\,a^2 \quad + \quad \ldots \quad + (k-2)\,a^{k-1} + \quad (k-1)\,a^k$$

restando la segunda ecuación a la primera (en la parte derecha se han emparejado los términos con el mismo exponente de manera que al restarlos desaparece la componente aritmética):

$$S - a\,S = \qquad\qquad a \quad + \quad a^2 \quad + \quad \ldots \quad + \quad a^{k-1} \quad - \ (k-1) \cdot a^k$$

Los $k-1$ primeros elementos forman una serie geométrica de razón a. El último elemento se trata aparte:

$$S = \frac{\sum_{j=1}^{k-1}(a^j) - (k-1)a^k}{1-a}$$

operando y simplificando:

$$= \frac{a\frac{1-a^{k-1}}{1-a} - (k-1)a^k}{1-a}$$

$$= \frac{\frac{a-a^k}{1-a} - (k-1)a^k}{1-a}$$

$$= \frac{(a-a^k) - (1-a)(k-1)a^k}{(1-a)^2}$$

simplificando un poco más:

$$S = \frac{a - ka^k + aka^k - aa^k}{(1-a)^2} \tag{7}$$

Con los dos sumatorios resueltos, sustituimos en la expresión 5, los resultados 6 y 7:

$$T(n) = a^k T\left(\frac{n}{a^k}\right) + \log_a(n)\frac{1-a^k}{1-a} - \frac{a - ka^k + aka^k - aa^k}{(1-a)^2}$$

Para que la recursión alcance el caso base, se ha de cumplir $a^k = n$; en consecuencia, también podemos hacer las sustituciones $k = \log_a n$ y $T(n/a^k) = 1$. Tenemos:

$$T(n) = n + \log_a(n)\frac{1-n}{1-a} - \frac{a - n\log_a n + an\log_a n - an}{(1-a)^2}$$

operando y simplificando:

$$\begin{aligned}
&= \frac{(1-a)^2 n + (1-a)(\log_a n - n\log_a n) - a + n\log_a n - an\log_a n + an}{(1-a)^2} \\
&= \frac{(1-a)^2 n + \log_a n - \cancel{n\log_a n} - a\log_a n + \cancel{an\log_a n} - a + \cancel{n\log_a n} - \cancel{an\log_a n} + an}{(1-a)^2} \\
&= \frac{(1-a)^2 n + \log_a n - a\log_a n - a + an}{(1-a)^2} \\
&= \frac{1}{(1-a)^2}\left((a^2 - a + 1)n + (1-a)\log_a n - a\right)
\end{aligned}$$

Por lo tanto, podemos concluir:

$$T(n) \in \Theta(n)$$

En el caso frecuente $a = 2$, tenemos:

$$T(n) = 3n - \log_2 n - 2$$

Ejercicio 5.12. *(difícil) Resolver*

$$T(n) = \begin{cases} 1 & n = 0 \\ n + 2T(n-1) & n > 0 \end{cases}$$

Solución. Desplegando:

$$\begin{aligned}
T(n) &\overset{1}{=} n + 2T\left(n-1\right) \\
&\overset{2}{=} n + 2\left[(n-1) + 2T\left(n-2\right)\right] = n + 2(n-1) + 2^2 T(n-2) \\
&\overset{3}{=} n + 2(n-1) + 2^2\left[(n-2) + 2T(n-3)\right] = n + 2(n-1) + 2^2(n-2) + 2^3 T(n-3) \\
&\dots \\
&\overset{k}{=} \sum_{i=0}^{k-1} 2^i (n-i) + 2^k T(n-k)
\end{aligned}$$

Pararemos cuando $n - k = 0$, o sea cuando $k = n$

$$\stackrel{n}{=} \sum_{i=0}^{n-1} 2^i (n-i) + 2^n$$

Por el ejercicio 3.3 sabemos que

$$\sum_{i=0}^{n} 2^i (n-i) = 2^{n+1} - 2 - n$$

Nótese que

$$\sum_{i=0}^{n} 2^i (n-i) = \sum_{i=0}^{n-1} 2^i (n-i)$$

Ya que cuando $i = n$ el segundo término se anula.

Por tanto,

$$T(n) = \left(2^{n+1} - 2 - n\right) + 2^n = 3 \cdot 2^n - 2 - n \in O(2^n)$$

6. Cálculo de complejidades

Ejercicio 6.1. *Realiza un estudio* ***analítico*** *de la complejidad temporal de la siguientes función en C++.*

```
1  int ejercicio1 (vector < int > & v){
2  
3      int n = v.size();
4      if (n > 0){
5          int j = n;
6          int sum = 0;
7          while (j > 0 and sum < 100){
8              j = j/2;
9              sum = 0;
10             for (int i = j; i < n; i++)
11                 sum+=v[i];
12         };
13         return j;
14     }
15     else return -1;
16 }
```

En el supuesto de que existan los casos mejor y peor identifica las instancias que pertenecen a cada caso y obtén las correspondientes cotas de complejidad.

Solución.

- **Tamaño del problema:**

  El tamaño del problema viene dado por el número de elementos del vector `v` (que se almacena en la variable `n`)

- **¿Existe mejor y peor caso diferenciados?**

  Sí, dependiendo del **contenido** del vector `v`, el `while` de la línea 7 se repetirá más o menos veces. Si después de ejecutar el `for` la variable `sum` es mayor o igual a 100, habrá una salida del while inmediata. En otro caso el `while` se ejecutará del orden de $\log_2 n$ veces.

- **Complejidad temporal en el mejor de los casos:**

  El mejor caso se dará cuando el bucle `while` termine prematuramente porque `sum < 100`. Eso se dará cuando los elementos del vector `v`, desde la mitad en adelante, sumen al menos 100.

  Es decir, en el mejor de los casos están todos los vectores `v`, de cualquier tamaño `n`, tal que $\sum_{i=\frac{n}{2}}^{n-1} v_i \geq 100$.

  En este caso, el `while` solo se ejecuta una vez y la complejidad vendrá dada por el número de veces que se repite el `for`.

$$C_{\min}(n) = \sum_{i=\frac{n}{2}}^{n-1} 1 = \sum_{i=0}^{n-1} 1 - \sum_{i=0}^{\frac{n}{2}-1} 1 = n - \frac{n}{2} \in \Omega(n)$$

- **Complejidad temporal en el peor de los casos:**

  El peor de los caso se dará cuando el bucle `while` nunca termine porque `sum < 100`, o sea, terminará cuando `j == 0`. En ese caso, la variable `j`, que en un principio vale `n`, irá dividiéndose entre dos hasta que valga cero y termine el bucle (nótese que la división es entera, por lo que `1/2 == 0`). Esto es, `j` irá tomando valores $n, \frac{n}{2}, \frac{n}{4}, \ldots, 1, 0$.

  Esto se puede representar en una tabla:

Iteración bucle `while`	**Valor de** `j` **al empezar el** `while`	**Pasos en iteración**
1	n	$\frac{n}{2}$
2	$\frac{n}{2}$	$\frac{n}{2} + \frac{n}{4}$
3	$\frac{n}{4}$	$\frac{n}{2} + \frac{n}{4} + \frac{n}{8}$
...	...	...
k	$\frac{n}{2^{k-1}} = 1$	$\sum_{j=1}^{k} \frac{n}{2^j}$

  Despejando de $\frac{n}{2^{k-1}} = 1$ podemos ver que el bucle se repetirá $k = \lfloor \log_2 n \rfloor + 1$ veces.

  Así que el número de pasos será:

$$C_{\max}(n) = \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} \sum_{j=1}^{k} \frac{n}{2^j} = n \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} \sum_{j=1}^{k} \frac{1}{2^j}$$

  Y como $\sum_{j=1}^{k} \frac{1}{2^j} \in \Theta(1)$

  Tenemos que:

$$C_{\max}(n) = n \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} \Theta(1) \in O(n \log(n)) \quad (8)$$

  Este ejercicio también podría haberse hecho dándose cuenta que:

Iteración bucle `while`	**Valor de** `j` **al empezar el** `while`	**Pasos en iteración**
1	n	$\frac{n}{2} = n - \frac{n}{2}$
2	$\frac{n}{2}$	$\frac{n}{2} + \frac{n}{4} = n - \frac{n}{4}$
3	$\frac{n}{4}$	$\frac{n}{2} + \frac{n}{4} + \frac{n}{8} = n - \frac{n}{8}$
...	...	...
k	$\frac{n}{k^{n-1}} = 1$	$\sum_{j=1}^{k} \frac{n}{2^j} = n - \frac{n}{2^k}$

Y el número de pasos será:

$$
\begin{aligned}
C_{\max}(n) &= \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} \left(n - \frac{n}{2^k} \right) \\
&= n \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} \left(1 - \frac{1}{2^k} \right) \\
&< n \sum_{k=1}^{\lfloor \log_2 n \rfloor + 1} 1 \in O(n \log(n))
\end{aligned}
$$

Ejercicio 6.2. *Realiza un estudio* ***analítico*** *de la complejidad temporal de la siguientes función en C++.*

```cpp
float Mochila(
  const vector <float> &v,
  const vector <unsigned> &p,
  unsigned P,
  int i
) {
  if( i >= 0 ){
    float S1;
    if( p[i] <= P )
      S1 = v[i] + Mochila( v, p, P-p[i], i-1 );
    else
      S1 = 0;
    float S2 = Mochila( v, p, P, i-1 );
    return max( S1, S2 );
  }
  return 0;
}
```

En el supuesto de que existan los casos mejor y peor identifica las instancias que pertenecen a cada caso y obtén las correspondientes cotas de complejidad.

Solución.

- **Tamaño del problema:** El tamaño del problema **no** viene dado por el tamaño de los vectores `v` o `p`, sino por el número de sus elementos que se usan. Observad que en cada llamada recursiva el parámetro `i` se decrementa en uno, y el caso base es cuando `i == 0`. Eso nos indica que se visitarán todos los elementos de `p` y de `v` menores o igual a `i`, o sea, que `i` es una buena medida del tamaño del problema. Aunque no es imprescindible, vamos a utilizar `n = i-1` como tamaño del problema para evitar tamaños negativos en el caso base.

- **¿Existe mejor y peor caso diferenciados?** Según lo que valga la expresión `p[i] <= P` puede ser que se ejecute una llamada recursiva o dos (obsérvese que esta expresión depende del **contenido** del vector `p`). Esto puede dar lugar a diferencia en la complejidad.

- **Complejidad temporal en el mejor de los casos:** El mejor de los casos se dará cuando siempre la expresión `p[i] <= P` se evalúe a `false`. En ese caso solo habrá una llamada recursiva y la relación de recurrencia correspondiente será:

  $$T(n) = \begin{cases} 1 & n = 0 \\ 1 + T(n-1) & n > 0 \end{cases}$$

  Esta ecuación de recurrencia es muy similar a la resuelta en el ejercicio 5.6, nótese que no cambia nada por terminar en 0 en vez de 1.

  Por tanto:

  $$T(n) \in \Omega(n)$$

- **Complejidad temporal en el peor de los casos:** El peor caso se dará cuando siempre se evalúe la expresión `p[i] <= P` a `true`. En ese caso siempre se realizarán dos llamadas recursivas.

  Afortunadamente las dos llamadas recursivas son a problemas del mismo tamaño, con lo que podemos agruparlas en la relación de recurrencia:

  $$T(n) = \begin{cases} 1 & n = 0 \\ 1 + 2T(n-1) & n > 0 \end{cases}$$

  Esta ecuación de recurrencia es muy similar a la resuelta en el ejercicio 5.7. Al igual que en el caso anterior, nótese que no cambia nada por terminar en 0 en vez de 1. Por tanto:

  $$T(n) \in O(2^n)$$