Combinatoria y Recurrencia

Mercè Claverol
Ester Simó
Marisa Zaragozá

Departamento Matemática Aplicada IV
EPSEVG - UPC

Índice general

Capítulo 1

Combinatoria

El objetivo de estudio de los próximos días de clase será introducir técnicas que nos permitan determinar el cardinal de un conjunto finito.

1.1. El Cardinal de un Conjunto

¿Qué significa que un conjunto tenga n elementos? Una forma de contestar a la pregunta es recordar cómo contamos conjuntos sencillos. Decimos las palabras un, dos, tres, etc., y señalamos cada vez un objeto. Cuando cada objeto ha recibido un número nos detenemos, y el último valor pronunciado es el número de elementos del conjunto. Si queremos traducir esta técnica de numerar al lenguaje matemático, debemos definir, para cada entero positivo n, el conjunto

$$N_n = \{1, 2, 3, \cdots, n\}$$

La técnica de numerar asigna a cada elemento de N_n uno del conjunto S que queremos contar; en otras palabras, determina una función f de N_n en S. Además está claro que la función es biyectiva, puesto que si hemos contado correctamente, cada elemento de S recibe exactamente una imagen. Así pues, si S es un conjunto, n un número entero positivo, y existe una biyección entre N_n y S, decimos que S tiene n elementos.

Definición 1.1

Si S tiene n elementos, escribiremos $|S| = n$ y diremos que el *cardinal* (o *tamaño*) de S es n. Para el conjunto vacío, $\emptyset$, tenemos una definición especial pero razonable,

$$|\emptyset| = 0.$$

Si $|S| = n$, a menudo escribiremos

$$S = \{s_1, s_2, \cdots, s_n\},$$

lo que realmente es otra forma de decir que existe una biyección $f : N_n \to S$ tal que $f(i) = s_i$ $(1 \leq i \leq n)$.

1.2. Utilizando Funciones para Determinar Cardinales

A continuación presentaremos una técnica que nos permitirá acotar/determinar el cardinal de un conjunto probando que los elementos que contiene pueden estar relacionados de algún modo con los elementos de otro conjunto de cardinal conocido. Para establecer relaciones entre conjuntos utilizaremos funciones.

Teorema 1.1 *Si A y B son conjuntos finitos y $f : A \to B$ es una función inyectiva, entonces $|A| \leq |B|$.*

Dem. Supongamos que A es el conjunto de n elementos $\{a_1, a_2, \cdots, a_n\}$. Dado que f aplica A en B, el conjunto $\{f(a_1), f(a_2), \cdots, f(a_n)\}$ es un subconjunto de B. Por ser f una aplicación inyectiva, todos los elementos $f(a_i)$ son diferentes y, por lo tanto, B también contiene al menos n elementos. Así, $|A| \leq |B|$. $\square$

Teorema 1.2 *Si A y B son conjuntos finitos y $f : A \to B$ es una función exhaustiva, entonces $|A| \geq |B|$.*

Dem. Supongamos que A es el conjunto de n elementos $\{a_1, a_2, \cdots, a_n\}$ y B es el conjunto de m elementos $\{b_1, b_2, \cdots, b_m\}$. Por definición de función, tendremos que todos los elementos de A tienen una única imagen. Por ser f una aplicación exhaustiva, tendremos que $\forall b_i,\ \exists a_i \in A_i,\ f(a_i) = b_i$. Para todo $b_i \in B$, elegiremos uno de los posibles elementos $a_i \in A$ tal que $f(a_i) = b_i$. Y como todos los elementos a_i son diferentes, tendremos que $|A| \geq |B|$.$\square$

Estos dos teoremas tienen un importante corolario:

Corolario 1.3 (Principio de Biyección) *Si A y B son conjuntos finitos y $f : A \to B$ es una función biyectiva, entonces $|A| = |B|$.*

Dem. Una función biyectiva, es inyectiva y exhaustiva a la vez. Por tanto, aplicando los dos teoremas anteriores se demuestra este resultado. $\square$

Este último resultado es muy simple y bastante potente. Supongamos que queremos conocer la cardinalidad de un conjunto A y que ya conocemos la cardinalidad del conjunto B. Si podemos encontrar una biyección entre los conjuntos A y B, entonces por el Principio de Biyección tendremos que $|A| = |B|$. Veamos un ejemplo en el cual lo aplicamos.

Ejemplo 1.1

Supongamos que tenemos un conjunto con n elementos, ¿cúantos subconjuntos tiene?

La siguiente definición nos permitirá reformular la cuestión anterior en términos del cardinal de un conjunto.

Definición 1.2

Definimos el *conjunto de las partes* de un conjunto S, denotado 2^S, como el conjunto de subconjuntos de S, es decir,

$$2^S = \{A / A \subseteq S\}.$$

(Por ejemplo, si $S = \{s_1, s_2\}$, entonces $2^S = \{\emptyset, \{s_1\}, \{s_2\}, \{s_1, s_2\}\}$. En este caso, S tiene cuatro subconjuntos y, equivalentemente, el conjunto potencia de S, 2^S, tiene cuatro elementos.)

Por tanto, nuestro problema será determinar el cardinal del conjunto potencia de un conjunto de n elementos. El método que aplicaremos para resolverlo será establecer una biyección entre el conjunto potencia y el conjunto de cadenas binarias de longitud n. Dado que hay 2^n cadenas binarias de longitud n (resultado que demostramos en el Ejemplo 2.4), tendremos que el cardinal del conjunto potencia es 2^n. Veamos los detalles de la demostración.

Sea S un conjunto de cardinal n al cual se le supone un orden

$$S = \{s_1, s_2, \cdots, s_n\},$$

y sea $B_n = \{b_1b_2\cdots b_n / b_i \in \{0,1\}\}$ el conjunto de cadenas binarias de longitud n.

Consideremos la función $f : 2^S \to B_n$ definida por $f(A) = b_1b_2\cdots b_n$ donde:

$$b_i = \begin{cases} 1 & \text{si } s_i \in A \\ 0 & \text{en otro caso.} \end{cases}$$

(Por ejemplo, para $S = \{s_1, s_2\}$ y $A = \{s_1\}$, entonces $f(A) = 10$). Veamos que esta función es una biyección:

1. *f está bien definida*, ya que cada elemento del conjunto 2^S está aplicado a un sólo elemento del conjunto B_n; es decir, a cada subconjunto de S se le asocia una única cadena binaria de longitud n.

2. *Veamos que f es inyectiva*, esto es, $\forall A_1, A_2 \in 2^S$,

   $$\text{si} \quad f(A_1) = f(A_2) = b_1b_2\cdots b_n, \ \text{ entonces } \ A_1 = A_2.$$

   Esto es cierto porque $s_i \in A_1 \leftrightarrow b_i = 1 \leftrightarrow s_i \in A_2$. Es decir, cada elemento de A_1 también está en A_2 y viceversa.

3. *Veamos que f es exhaustiva*, esto es, para todo $b_1b_2\cdots b_n \in B_n$, existe almenos un elemento $A \in 2^S$ tal que $f(A) = b_1b_2\cdots b_n$.

   Para probarlo, construimos A explícitamente:

   $$A = \{s_i \ / \ b_i = 1, 1 \leq i \leq n\}$$

   Entonces $f(A) = b_1b_2\cdots b_n$ como queríamos.

Por tanto, concluimos que f es biyectiva y por el Principio de Biyección los dos conjuntos tienen el mismo cardinal:

$$|2^S| = |B_n| = 2^n.$$

1.2.1. El principio de las Cajas

El Teorema 2.1 tiene otro importante corolario:

Corolario 1.4 (Principio de las Cajas) *Si $k+1$ o más objetos se distribuyen en k cajas, hay al menos una caja que contiene dos o más objetos.*

Dem. Denotemos por A al conjunto de los objetos y B al conjunto de las cajas.

Para demostrar este resultado, procederemos por reducción al absurdo y *supondremos que ninguna de las cajas contiene más de un objeto.*

Si establecemos una aplicación entre los conjuntos A y B de modo que a cada objeto se le asocia la caja que le contiene, la suposición anterior implica que la función es inyectiva. Entonces, aplicando el Teorema 2.1 tenemos que

$$k+1 \leq |A| \leq |B| = k,$$

lo cual es absurdo. □

Veamos un ejemplo en el cual aplicaremos este resultado.

Ejemplo 1.2

Una red de ordenadores consta de 6 ordenadores. Cada uno de ellos está conectado directamente a 0 o más de los otros ordenadores. Probar que hay al menos 2 en la red conectados al mismo número de ordenadores.

Dado que hay 6 ordenadores en la red, el número de conexiones de un ordenador es un valor entre 0 y 5.

Denotamos por A al conjunto de ordenadores y establecemos una aplicación f entre A y el conjunto $B = \{0, 1, 2, 3, 4, 5\}$, de modo que a cada ordenador se le asocia su número de conexiones.

Notemos que no puede haber simultáneamente en la red un ordenador con 0 conexiones y otro con 5 conexiones. Por tanto, el cardinal del conjunto imagen $f(A)$ es menor o igual a 5.

Por lo tanto, tenemos 6 ordenadores (6 objetos) para asignarles el número de conexiones que tendrán en la red (5 cajas $\equiv$ 5 números posibles de conexión) y, aplicando el Principio de las Cajas, habrán al menos dos ordenadores con el mismo número de conexiones.

A continuación presentaremos la generalización del principio anterior.

Teorema 1.5 (Principio de las Cajas Generalizado) *Si n objetos se distribuyen en k cajas, hay al menos una caja conteniendo $\lceil \frac{n}{k} \rceil$ o más objetos.*

Dem. Supongamos que ninguna de las cajas contiene más de $\lceil \frac{n}{k} \rceil$-1 objetos. Entonces, el número total de objetos es como mucho

$$k \cdot (\lceil \frac{n}{k} \rceil - 1) < k \cdot ((\frac{n}{k} + 1) - 1) = n$$

Esto contradice el hecho de que hayan n objetos. □

1.3. La Cardinalidad de la Unión de Conjuntos

Continuaremos nuestro estudio sobre las técnicas de enumeración con dos principios básicos del conteo: La Regla de la Suma y el Principio de Inclusión-Exclusión.

Supongamos que conocemos el tamaño de algunos conjuntos $A_1, A_2, \cdots, A_n$ y nos preguntamos ¿cuántos elementos hay en la unión? Si los conjuntos son disjuntos, el tamaño de la unión viene dado por la *Regla de la Suma*. Si los conjuntos no son necesariamente disjuntos, entonces el tamaño vendrá dado por el *Principio de Inclusión-Exclusión*.

1.3.1. La Regla de la Suma

Teorema 1.6 *Si $A_1, A_2, \cdots, A_n$ son conjuntos disjuntos, entonces:*

$$|A_1 \cup A_2 \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|.$$

La Regla de la Suma dice que el número de elementos en una unión de conjuntos disjuntos es igual a la suma de los tamaños de todos los conjuntos. Esta regla se puede demostrar por inducción sobre el número de conjuntos (dejaremos la demostración como ejercicio para el lector).

Como observaremos en el ejemplo siguiente, la regla de la suma subyace en algunos de los cálculos elementales que suelen efectuarse, sin necesidad de pensar qué principio es apropiado aplicar.

Ejemplo 1.3

Cuando se utiliza cierto procesador de textos en un ordenador personal, se tiene la opción de utilizar un menú llamado *file* y un menú llamado *edit*. Si se elige el menú *file*, se proporcionan siete actividades: nuevo, abrir, cerrar, guardar, guardar como, imprimir y salir. Y si se elige el menú *edit*, se tienen ocho opciones: deshacer, cortar, copiar, pegar, eliminar, buscar, reemplazar y selecionar todo. ¿Cuántas actividades diferentes es posible solicitar con los dos menús?

Se pueden elegir siete actividades de archivo y ocho de edición. Los conjuntos de actividades de archivo y edición no tienen nada en común, por lo que el número total de actividades que se podrán realizar utilizando ambos menús es 7+8=15.

1.3.2. El Principio de Inclusión-Exclusión

Así como la Regla de la Suma nos permite calcular la cardinalidad de una unión de conjuntos disjuntos, el Principio de Inclusión-Exclusión da la cardinalidad de una unión de conjuntos que pueden interseccionar. Empezaremos con los casos especiales $n = 2$ y $n = 3$.

El Principio de Inclusión-Exclusión para dos subconjuntos

Teorema 1.7 *Sean A y B dos subconjuntos, no necesariamente disjuntos,*

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Dem. Los elementos de la intersección se cuentan dos veces en $|A| + |B|$, por lo tanto, para contar cada elemento de la unión sólo una vez, restamos $|A \cap B|$ a la expresión anterior. □

El Principio de Inclusión-Exclusión para tres subconjuntos

Teorema 1.8 *Sean A, B y C tres subconjuntos, no necesariamente disjuntos,*

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Dem. Los elementos que pertenecen sólo a un conjunto se cuentan una vez en $|A| + |B| + |C|$. Dado que estos elementos no están incluidos en ninguna intersección de conjuntos, no se restarán en $-|A \cap B| - |A \cap C| - |B \cap C|$, ni se sumarán en $|A \cap B \cap C|$.

Los elementos contenidos en dos de los conjuntos A, B y C se cuentan dos veces en $|A|+|B|+|C|$, se restan una vez en $-|A \cap B|-|A \cap C|-|B \cap C|$, y no se vuelven a contar en $|A \cap B \cap C|$ al no estar incluidos en la intersección de los tres conjuntos. Por lo tanto, estos elementos se cuentan 2+1=1 vez en la fórmula anterior

Los elementos incluidos en los tres conjuntos A, B y C, se cuentan tres veces en $|A|+|B|+|C|$, se restan de nuevo tres veces en $-|A\cap B|-|A\cap C|-|B\cap C|$, y se añaden una vez en $|A\cap B\cap C|$. Por lo tanto, estos elementos se cuentan una vez (3-3+1=1) en la fórmula anterior. $\square$

El nombre de *Inclusión-Exclusión* viene del modo en que los elementos son contados, restados, vueltos a contar, etc.

El Principio de Inclusión-Exclusión (caso general)

Teorema 1.9 *Sean $A_1, A_2, \cdots, A_n$ conjuntos, no necesariamente disjuntos. La cardinalidad de la unión se calcula como sigue:*

$$\begin{array}{rcl} |A_1\cup A_2\cup\cdots A_n| & = & |A_1|+|A_2|+\cdots+|A_n| \\ & - & \sum |A_i\cap A_j| \\ & + & \sum |A_i\cap A_j\cap A_k| \\ & & \vdots \\ & + & (-1)^{n+1}|A_1\cap A_2\cap\cdots\cap A_n| \end{array}$$

El Principio de Inclusión-Exclusión se utiliza para calcular el cardinal de un conjunto que cumple una serie de propiedades. Veamos un ejemplo.

Ejemplo 1.4 (Contando Números Primos)

¿Cuántos de los números 1, 2, $\cdots$, 100 son primos? Para hallar el número de primos en este rango, primero contaremos el número de enteros compuestos que no excedan a 100 y restaremos este valor a 99 (porque el 1 no se considera ni primo ni compuesto).

Recordemos que un entero compuesto es divisible por un número primo que no exceda su raiz cuadrada. Por lo tanto, los enteros compuestos que no excedan 100 deben tener un factor primo que no exceda a 10, a saber 2, 3, 5 y 7.

Si definimos por A_i el subconjunto de números enteros en el rango $1, 2, \cdots, 100$, que son divisibles por i, siendo $i = 2$, 3, 5, 7, tendremos que el número de enteros compuestos en este rango vendrá dado por la expresión

$$|A_2\cup A_3\cup A_5\cup A_7|$$

Aplicando el Principio de Inclusión-Exclusión obtendremos

$$\begin{aligned}|A_2 \cup A_3 \cup A_5 \cup A_7| \quad &= |A_2| + |A_3| + |A_5| + |A_7| \\ &-|A_2 \cap A_3| - |A_2 \cap A_5| - |A_2 \cap A_7| \\ &-|A_3 \cap A_5| - |A_3 \cap A_7| - |A_5 \cap A_7| \\ &+|A_2 \cap A_3 \cap A_5| + |A_2 \cap A_3 \cap A_7| \\ &+|A_2 \cap A_5 \cap A_7| + |A_3 \cap A_5 \cap A_7| \\ &-|A_2 \cap A_3 \cap A_5 \cap A_7|\end{aligned}$$

Para calcular estos cardinales, notemos que el número de enteros entre 2 y 100 que son divisibles por cualquier subconjunto de primos del conjunto $\{2, 3, 5, 7\}$ es $\lfloor \frac{100}{N} \rfloor$, donde N denota al producto de los primos del subconjunto, (esto se debe a que cualquier par de números primos de este subconjunto, no tienen factores en común). Por tanto,

$$\begin{aligned}|A_2 \cup A_3 \cup A_5 \cup A_7| \quad &= 50 + 33 + 20 + 14 \\ &-16 - 10 - 7 - 6 - 4 - 2 \\ &+3 + 2 + 1 + 0 \\ &-0 \\ &= 78.\end{aligned}$$

Por tanto, el número de primos en este rango es

$$\begin{aligned}&|\overline{A_2 \cup A_3 \cup A_5 \cup A_7}| + |\{2, 3, 5, 7\}| \\ &= 99 - |A_2 \cup A_3 \cup A_5 \cup A_7| + 4 = 99 - 78 + 4 = 25.\end{aligned}$$

1.4. Producto de Conjuntos

Recordemos la definición del producto de dos conjuntos:

$$A \times B = \{(a, b) \,/\, a \in A,\, b \in B\}$$

(Por ejemplo, Si $A = \{x, y, z\}$ y $B = \{1, 2\}$, entonces

$$A \times B = \{(x, 1), (x, 2), (y, 1), (y, 2), (z, 1), (z, 2)\}.$$

En este caso, A contiene 3 elementos, B contiene 2 elementos y el producto contiene $3 \cdot 2 = 6$ elementos.)

De forma general, el tamaño de un producto de conjuntos viene dado por la *Regla del Producto*:

Teorema 1.10 (Regla del Producto) *Si $A_1, A_2, \cdots, A_n$ son conjuntos, entonces:*

$$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \cdot |A_2| \cdots |A_n|.$$

La demostración es por inducción sobre n, pero no la haremos aquí.

Otra forma de interpretar la regla del producto sería: El número de formas de elegir un elemento de A_1, otro elemento de A_2, $\cdots$, y un elemento de A_n es $|A_1| \cdot |A_2| \cdots |A_n|$.

Ejemplo 1.5 (Cadenas binarias)

¿Cuántas cadenas binarias de n-dígitos hay? Si denotamos por $B = \{0, 1\}$, entonces el conjunto de cadenas binarias de n dígitos es:

$$B \times B \times \cdots \times B \qquad (n \text{ términos})$$

Por la Regla del Producto, el número de cadenas binarias de n dígitos es $|B|^n = 2^n$

Ejemplo 1.6 (Passwords)

Supongamos que un password consta de 8 caracteres dónde cada caracter es o un número o una letra minúscula. Un password es *legal* si hay al menos un número y al menos una letra. ¿Cuántos passwords legales habrá?

Podemos resolver este problema contando los elementos del conjunto complementario. Calcularemos el número de passwords legales, a partir del número de passwords ilegales. Un password ilegal tiene o todo números o todo letras. Por la Regla del Producto, la cantidad de passwords con todo letras es 26^8, y el número de passwords con todo números es 10^8. También, por la Regla del Producto, el número de passwords posibles (tanto legales como ilegales) es 36^8. Por lo tanto, el número total de passwords legales es $36^8 - 10^8 - 26^8$.

A veces es necesario combinar varios tipos diferentes de principios de conteo en la solución de un problema. En el siguiente ejemplo veremos que necesitamos ambas reglas, la de la Suma y la del Producto, para obtener el resultado.

Ejemplo 1.7 (Nombre de variables)

En algunas de las primeras versiones del lenguaje de programación BASIC, el nombre de una variable constaba de una sola letra (A, B, C, ...) o una sola letra seguida de un solo número. Como el ordenador no distinguía entre las letras mayúsculas y minúsculas, a y A se consideraban como el mismo nombre de variable, así como también e7 y E7. Aplicando la Regla del Producto, tenemos que existían $26 \cdot 10 = 260$ nombres de variables que constaban de una letra seguida por un número; y como habían 26 nombres de variables que constaban de una sola letra, por la Regla de la Suma tenemos que existían 260+26=286 nombres posibles de variables en ese lenguaje de programación.

1.5. Permutaciones

Las permutaciones son maneras de distribuir objetos.

Definición 1.3

Una *permutación de m elementos* de un subconjunto S de cardinal n, con $m \leq n$, se define como una función inyectiva del conjunto $M = \{1, 2, \cdots, m\}$ al conjunto S. Los valores de una función de este tipo determinan una $m-$pla

$$(f(1), f(2), \cdots, f(m))$$

de elementos diferentes de S,

$$f(1) = s_1, f(2) = s_2, \cdots, f(m) = s_m$$

Así, una permutación de m elementos de S es una selección ordenada (lista) de m elementos distintos de S, que denotaremos $s_1 s_2 \cdots s_m$. Por ejemplo, la terna (c, a, b) corresponde a la función f definida por

$$f(1) = c, f(2) = a, f(3) = b,$$

y a la permutación *cab*.

A continuación enumeraremos todas las permutaciones de 3 elementos del conjunto $S = \{a, b, c, d\}$. Las permutaciones las escribiremos en orden alfabético.

$$\begin{array}{cccccc} abc & abd & acb & acd & adb & adc \\ bac & bad & bca & bcd & bda & bdc \\ cab & cad & cba & cdb & cda & cdb \\ dab & dac & dba & dbc & dca & dcb \end{array}$$

Una permutación de n elementos de un conjunto S de n elementos la denominaremos simplemente *permutación* de S. Enumeremos todas las permutaciones del conjunto $\{a, b, c\}$.

$$abc \quad acb \quad bac \quad bca \quad cab \quad cba$$

1.5.1. Contando Permutaciones

Para muchos propósitos es importante contar el número de permutaciones.

Teorema 1.11 *El número de permutaciones de m elementos de un conjunto S de cardinal n es*

$$P(n, m) = \frac{n!}{(n-m)!}$$

Dem. Cada aplicación inyectiva de $M = \{1, 2, \cdots, m\}$ en S queda determinada de forma única por la selección ordenada de los valores distintos $s_1 s_2 \cdots s_m$. La primera selección puede ser cualquiera de los n objetos de S. Puesto que no están permitidas repeticiones, la segunda selección debe ser uno de los $n-1$ objetos restantes. Igualmente, hay $n-2$ posibilidades para s_3, y así sucesivamente. Cuando llegamos a la selección de s_m, ya han sido seleccionados m-1 objetos, de forma que s_m ha de ser uno de los $n-(m-1)$ elementos restantes. Por el Principio del Producto,

$$\begin{aligned} P(n, m) &= n \cdot (n-1) \cdots (n-(n-1)) \\ &= n \cdot (n-1) \cdots (n-(n-1)) \tfrac{(n-m)(n-(m+1))\cdots 2\cdot 1}{(n-m)(n-(m+1))\cdots 2\cdot 1} \\ &= \tfrac{n!}{(n-m)!} \end{aligned}$$

$\square$

1.6. Permutaciones con Repetición

Ejemplo 1.8

Un pequeño sistema de manejo de base de datos consta de tres dispositivos de memoria y cuatro estaciones de trabajo. En cualquier instante, cualquier estación puede hacer una petición de datos a cualquier dispositivo de memoria. ¿Cuántos patrones distintos para peticiones simultáneas de todas las estaciones de trabajo son posibles?

Un patrón puede considerarse como una cuádrupla integrada por el dispositivo solicitado por la estación 1, el dispositivo solicitado por la estación 2, el dispositivo solicitado por la estación 3 y el dispositivo solicitado por la estación 4. Para la primera posición de la cuádrupla hay tres posibles elementos; para cada una de las formas de elegir el primer elemento de la cuádrupla hay tres formas de elegir el segundo elemento; para cada elección de los dos primeros elementos de la cuádrupla hay tres formas de elegir el tercer elemento; y para cada forma de elegir los tres primeros elementos hay tres opciones para elegir el cuarto elemento de la cuádrupla. Por tanto se tienen $3 \cdot 3 \cdot 3 \cdot 3 = 3^4$ cuádruplas: 3^4 patrones posibles de solicitar datos.

En el ejemplo anterior hemos visto otro caso especial de distribuciones que se presenta a menudo: las permutaciones con repetición.

Definición 1.4

Una *permutación con repetición de m elementos* de un conjunto S de cardinal n, con $n \leq m$, se define como una función del conjunto $M = \{1, 2, \cdots m\}$ al conjunto S.

1.6.1. Contando Permutaciones con Repetición

Afortunadamente, las permutaciones con repetición son fáciles de contar.

Teorema 1.12 *El número de permutaciones con repetición de m elementos de un conjunto S de cardinal n es n^m.*

Dem. Las funciones f pueden describirse como $m-$plas

$$(f(1), f(2), \cdots, f(m)).$$

Para cada $f(i)$ hay n elecciones, de modo que por el Principio del Producto hay n^m funciones diferentes. $\square$

1.6.2. Permutaciones con Repetición Limitada

Consideremos un caso especial de permutaciones con repetición: las permutaciones de m elementos donde cada elemento se repite un número específico de veces. Veamos un ejemplo.

Ejemplo 1.9

¿Cuál es el número de disposiciones de las seis letras de la palabra PEPPER? Esto es igual al número de permutaciones con repetición de 6 elementos del conjunto $\{P, E, R\}$ donde P se repite 3 veces, E se repite 2 veces, y R se repite una vez.

Inicialmente, supondremos que todas las letras son diferentes, para ello añadiremos subíndices. Esto es, queremos calcular el número de permutaciones de las letras $P_1E_1P_2P_3E_2R$. En este caso, hay $6! = 720$ disposiciones porque hay seis elecciones para la primera letra, cinco para la segunda, etc.

Existen $3! = 6$ disposiciones con las letras P para cada permutación en las que las letras P no se distinguen. Por ejemplo, $P_1E_1P_2P_3E_2R$, $P_1E_1P_3P_2E_2R$, $P_2E_1P_1P_3E_2R$, $P_2E_1P_3P_1E_2R$, $P_3E_1P_1P_2E_2R$ y $P_3E_1P_2P_1E_2R$ corresponden a PE_1PPE_2R cuando eliminamos los subíndices de las letras P. Y además, a la disposición $P_1EP_2P_3ER$ le corresponde la pareja de permutaciones $P_1E_1P_2P_3E_2R$, $P_1E_2P_2P_3E_1R$, cuando se distinguen las letras E. En consecuencia,

$$\begin{aligned} 2! \cdot 3! \cdot \ & N^{\text{o}} \text{ de disposiciones de las letras de } PEPPER \\ = \ & N^{\text{o}} \text{ de permutaciones de los símbolos } P_1, E_1, P_2, P_3, E_2, R. \end{aligned}$$

De la igualdad anterior, tendremos que el número de disposiciones de las letras de la palabra $PEPPER$ es $\frac{6!}{2!3!} = 60$.

Siguiendo el mismo argumento utilizado en el ejemplo anterior, podríamos demostrar el siguiente resultado. (La demostración se deja como ejercicio para el lector).

Teorema 1.13 *Sea S el conjunto $\{s_1, s_2, \cdots, s_n\}$, y sean $m_1, m_2, \cdots, m_n$ enteros no negativos. El número de permutaciones del conjunto S donde cada elemento s_i se repite m_i veces es:*

$$\frac{(m_1 + m_2 + \cdots + m_n)!}{m_1! m_2! \cdots m_n!}$$

1.7. Combinaciones

Es posible que una unidad de disco de ordenador utilice palabras de 16 bits (series de 16 ceros y unos) con información codificada de alguna manera especial. Con esta codificación especial, si cualquier bit tiene error, el ordenador lo corregirá. Además, si dos bits cualesquiera tienen error, el ordenador detecta esto y emite un mensaje comunicándolo. ¿Cuántos patrones de dos errores son posibles? Tal patrón de errores es un subconjunto de dos elementos del conjunto de 16 posiciones de la palabra, de modo que se está preguntando ¿cuántos subconjuntos de dos elementos tiene un conjunto de 16 elementos?

Para calcular el número de subconjuntos de m elementos de un conjunto de n elementos, en otras palabras, el número de *combinaciones* de m elementos de un conjunto de n elementos, aplicaremos de forma indirecta el Principio del Producto. El método se basa en relacionar permutaciones y combinaciones de m elementos.

Cada permutación de m elementos elegida de un conjunto S de cardinal n es una selección ordenada de m elementos distintos de S. Por consiguiente, se trata de un listado de los elementos de un subconjunto de S de cardinal m. Un subconjunto dado de m elementos puede ordenarse de $m!$ formas, de modo que por el Principio del Producto se obtiene:

$$m! \cdot C(n, m) = P(n, m)$$

siendo $C(n, m)$ el número de combinaciones de m elementos de un conjunto de n elementos y $P(n, m)$ el número de permutaciones de m elementos de un conjunto de n elementos. Y despejando de la igualdad anterior obtenemos

$$\begin{aligned} C(n, m) &= \tfrac{P(n,m)}{m!}, \\ &= \tfrac{n!}{(n-m)!m!}, \end{aligned}$$

Por tanto, acabamos de demostrar el siguiente teorema.

Teorema 1.14 *El número de combinaciones de m elementos de un conjunto S de cardinal n está dado por*

$$C(n,m) \quad = \tfrac{n!}{(n-m)!m!}.$$

Apliquémoslo para contestar a la pregunta que se hizo al principio de la sección.

Ejemplo 1.10

¿Cuál es el número de subconjuntos de dos elementos de un conjunto de 16 elementos? Este número es

$$C(16,2) = \frac{16!}{14!2!} = \frac{16 \cdot 15 \cdot 14!}{14!2!} = \frac{16 \cdot 15}{2} = 120.$$

Existe otra notación frecuente para $C(n,m)$, esta es $\binom{n}{m}$, que se lee *n sobre m* y se denomina *coeficiente/número binomial*, por razones que se explicarán después. En general, el cálculo de los números binomiales depende del siguiente resultado fundamental.

Teorema 1.15 *Si n y m son enteros positivos, con $1 \leq m \leq n$, entonces*

$$\tbinom{n}{m} = \tbinom{n-1}{m-1} + \tbinom{n-1}{m}$$

Dem. Sea S un conjunto de cardinal n y supongamos que x es cierto elemento de S. El conjunto de todos los subconjuntos de cardinal m del conjunto S puede dividirse en dos partes disjuntas U y V de la forma siguiente:

$$\begin{aligned} U &= \text{los subconjuntos de cardinal } m \text{ que contienen a } x, \\ V &= \text{los subconjuntos de cardinal } m \text{ que no contienen a } x. \end{aligned}$$

Un subconjunto de cardinal m es de U si (y sólo si), al quitarle x se obtiene un subconjunto de cardinal $m-1$ del conjunto $S - \{x\}$. Así pues,

$$|U| = \tbinom{n-1}{m-1}.$$

Por otra parte, un subconjunto de cardinal m es de V si (y sólo si) es un subconjunto de cardinal m del conjunto $S - \{x\}$. Por lo tanto,

$$|V| = \tbinom{n-1}{m}$$

Aplicando el Principio de la Suma tenemos que el número total de subconjuntos de cardinal m es igual a $|U| + |V|$ y, dado que este número es $\binom{n}{m}$, tenemos el resultado. $\square$

El teorema anterior proporciona un método recursivo para calcular los números binomiales, ya que si conocemos los números $\binom{n-1}{m}$ para $0 \leq m \leq n-1$, podemos calcular los coeficientes binomiales $\binom{n}{m}$. Estos cálculos se suelen mostrar en forma de triángulo (conocido como *Triángulo de Pascal*).

$$
\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1 \\
 & & & & & & & \vdots & & & & & & &
\end{array}
$$

1.7.1. El teorema del Binomio

Cuando se introdujeron los coeficientes binomiales, no se pudo explicar de donde venía su nombre. Los coeficientes binomiales obtienen su denominación de su relación con las potencias de un tipo especial de polinomio: el binomio. (Un binomio es un polinomio que es suma de dos términos)

Al aplicar las reglas aprendidas en álgebra para multiplicar productos polinomiales, podemos comprobar los cálculos siguientes:

$$
\begin{array}{rc}
(x+y)^0 = & 1 \\
(x+y)^1 = & x+y \\
(x+y)^2 = & x^2+2xy+y^2 \\
(x+y)^3 = & x^3+3x^2y+3xy^2+y^3 \\
(x+y)^4 = & x^4+4x^3y+6x^2y^2+4xy^3+y^4 \\
(x+y)^5 = & x^5+5x^4y+10x^3y^2+10x^2y^3+5xy^4+y^5
\end{array}
$$

Analicemos un caso particular. Si $n=4$, el coeficiente de x^2y^2 en el desarrollo del producto

$$
\begin{array}{cccc}
(x+y) & (x+y) & (x+y) & (x+y) \\
\text{primer} & \text{segundo} & \text{tercer} & \text{cuarto} \\
\text{factor} & \text{factor} & \text{factor} & \text{factor}
\end{array}
$$

es el número de formas en que podemos seleccionar dos de las cuatro x. (Observemos que aunque las x son iguales en apariencia, las distinguimos como la x del primer factor, la x del segundo factor, y así sucesivamente. También observamos que, cuando selecionamos dos x, usamos dos factores, lo que nos deja otros dos factores de los que podemos seleccionar las dos y necesarias.) Por ejemplo, entre las posibilidades, podemos seleccionar x de los dos primeros factores e y de los dos últimos, o x de los factores primero y tercero e y del segundo y el cuarto. La siguiente tabla resume las seis opciones posibles:

Factores seleccionados para x	Factores seleccionados para y
(1ª) 1,2	(1ª) 3,4
(2ª) 1,3	(2ª) 2,4
(3ª) 1,4	(3ª) 2,3
(4ª) 2,3	(4ª) 1,4
(5ª) 2,4	(5ª) 1,3
(6ª) 3,4	(6ª) 1,2

En consecuencia, el coeficiente de x^2y^2 en el desarrollo de $(x+y)^4$ es $\binom{4}{2} = 6$, el número de formas de elegir dos objetos distintos de una colección de cuatro objetos diferentes. Este resultado sugiere el siguiente teorema.

Teorema 1.16 (El Teorema del Binomio) *Si x e y son variables y n es un entero positivo, entonces*

$$\begin{aligned}(x+y)^n &= \tbinom{n}{0}x^0y^n + \tbinom{n}{1}x^1y^{n-1} + \tbinom{n}{2}x^2y^{n-2} + \cdots + \tbinom{n}{n-1}x^{n-1}y^1 + \tbinom{n}{n}x^ny^0 \\ &= \textstyle\sum_{k=0}^{n}\tbinom{n}{k}x^ky^{n-k}.\end{aligned}$$

Dem. Veamos que ocurre al multiplicar n factores

$(x+y)$	$(x+y)$	$(x+y)$	$\cdots$	$(x+y)$
primer factor	segundo factor	tercer factor		n-ésimo factor

Los términos del producto se obtienen seleccionando x o bien y de cada factor. El coeficiente de x^ky^{n-k} (El número de términos x^ky^{n-k}) es exactamente el número de formas distintas en que podemos elegir k veces x (y en consecuencia $n-k$ letras y) de los n factores disponibles, lo cual, por definición, es igual al número binomial $\binom{n}{k}$. $\square$

1.8. Combinaciones con Repetición

Sea el conjunto $S = \{A, B, C\}$. Tenemos tres formas de elegir dos elementos de S sin importarnos el orden, en otras palabras, este conjunto tiene 3 combinaciones de dos elementos, las cuales presentamos a continuación:

$$\{A, B\}\ \{A, C\}\ \{B, C\}$$

Si se permiten repeticiones, tenemos 6 selecciones no ordenadas de dos elementos del conjunto S, a saber:

$$\{A, B\}\ \{A, C\}\ \{B, C\}\ \{A, A\}\ \{B, B\}\ \{C, C\}$$

Demostraremos que es posible obtener una fórmula general para el número de selecciones no ordenadas con repetición (*combinaciones con repetición*) en términos de los números binomiales.

Teorema 1.17 *El número de combinaciones con repetición de m elementos de un conjunto de cardinal n es*

$$\binom{n+m-1}{m}$$

Dem. Sea S un conjunto con n elementos, el cual suponemos que está ordenado de algún modo. Estableceremos una biyección entre las combinaciones con repetición de m elementos del conjunto S y las palabras en el alfabeto $\{0, 1\}$.

Sea R una combinación con repetición de m elementos de S. Consideraremos $n - 1$ ceros. Estos ceros definen n regiones. Pondremos un uno en la $i-$ésima región, por cada vez que el $i-$ésimo elemento de S aparece en la combinación R. Este procedimiento aplica cada combinación con repetición a una palabra en el alfabeto $\{0, 1\}$.

(Por ejemplo, sea S el conjunto $\{A, B, C, D, E\}$, con los elementos ordenados alfabéticamente. Sea R=$\{A, B, B, B, D, E, E\}$ la combinación con repetición de 7 elementos de S . La palabra que se corresponde con R será:

$$\underbrace{1}_{A}\quad 0\quad \underbrace{1\,1\,1}_{B,B,B}\quad 0\ \ 0\quad \underbrace{1}_{D}\quad 0\quad \underbrace{1\,1}_{E,E}$$

Los dos ceros consecutivos indican que el elemento C no aparece en R.)

Esta aplicación es una biyección porque tiene inversa. Es decir, dada una palabra en el alfabeto $\{0, 1\}$ podemos construir la correspondiente combinación con repetición. El número de unos en la primera región determina el número de veces que el primer elemento de S aparece en la combinación con repetición, el número de unos en la segunda región determina el número de veces que aparece el segundo elemento, etc.

Dado que la aplicación anterior es una biyección, el número de combinaciones con repetición de m elementos de un conjunto de cardinal n es igual al número de palabras que contienen $n-1$ ceros y m unos. Los ceros pueden ocupar cualesquiera de las $n+m-1$ posiciones, de forma que el número de palabras es

$$\tbinom{n+m-1}{n-1} = \tbinom{n+m-1}{m}$$

como queríamos demostrar. □

Ejemplo 1.11

Consideremos el siguiente segmento de un programa en Pascal, donde i, j, k son variables enteras.

```
For i:= 1 to 20 do
    For j:=1 to i do
        For k:=j downto 1 do
          Writeln ((i*j)+k);
```

¿Cuántas veces se ejecuta la proposición Writeln en este segmento de programa?

Entre las opciones posibles de i, j y k que nos conducen a la ejecución de la proposición Writeln, enumeramos algunas

$$\begin{array}{llll}
(1) & i=1, & j=1, & k=1; \\
(2) & i=2, & j=1, & k=1; \\
(3) & i=15, & j=10, & k=1; \\
(4) & i=15, & j=10, & k=7.
\end{array}$$

Observamos que $i = 10, j = 12, k = 5$ no es una de las selecciones que deberán considerarse, ya que $j = 12 > 10 = i$, lo cual viola la condición establecida en el segundo ciclo For. Cada una de las cuatro selecciones anteriores en las que se ejecuta la proposición Writeln satisface la condición $1 \leq k \leq j \leq i \leq 20$. De hecho, cualquier selección a, b, c ($a \leq b \leq c$) de tamaño 3, con repeticiones, de la lista $1, 2, 3, \cdots, 20$ produce una de las selecciones correctas: en este caso $k = a, j = b, i = c$. En consecuencia, la proposición Writeln se ejecuta

$$\binom{20+3-1}{3} = \binom{22}{3} = 1540 \text{ veces.}$$

En el siguiente ejemplo se introduce una idea que parece tener más que ver con la teoría de números que con las combinaciones. Sin embargo, la solución de este ejemplo será equivalente al conteo de las combinaciones con repetición.

Ejemplo 1.12

Determine el número de soluciones enteras de la ecuación

$$x_1 + x_2 + x_3 + x_4 = 7, \quad \text{donde } x_i \geq 0 \text{ para toda } 1 \leq i \leq 4.$$

Una solución de la ecuación es $x_1 = 3, x_2 = 3, x_3 = 0, x_4 = 1$, la cual se podría interpretar como que estamos distribuyendo siete objetos idénticos (siete unos) en cuatro regiones diferentes, y que hemos colocado 3 unos en la primera región, 3 unos en la segunda, ninguno en la tercera y un uno en la cuarta región. Si seguimos con esta interpretación, vemos que cada solución entera no negativa de la ecuación se corresponde a una selección con repetición de tamaño 7 (los objetos idénticos) de un conjunto de cardinal 4 (las regiones diferentes), de modo que hay $\binom{4+7-1}{7} = 120$ soluciones enteras de la ecuación.

Capítulo 2

Recurrencia

2.1. Preliminares

Definición 2.1

Una sucesión numérica a_n, $n = 0, 1, 2, \cdots$ puede expresarse recursivamente si:

- se enuncian explícitamente los valores iniciales $a_0, a_1, \cdots, a_{c-1}$ (*condiciones iniciales*),
- y mientras que para todo $n \geq c$ se obtiene a_n a partir de la relación

  $$a_n = f(a_{n-1}, a_{n-2}, \cdots, a_{n-c}, n)$$

  esto es, una ecuación que permite calcular el n-ésimo término a partir de términos precedentes (las relaciones de este tipo se denominan *relaciones de recurrencia*).

Ejemplo 2.1

Supongamos que tenemos

$$\begin{aligned} &a_1 = 1, a_2 = 2, \\ &a_n = a_{n-1} + a_{n-2},\ \forall n \geq 3 \end{aligned}$$

La ecuación

$$a_n = a_{n-1} + a_{n-2} \quad \forall n \geq 3$$

indica la forma de obtener nuevos elementos de la sucesión a partir de los resultados anteriores ya conocidos (o que se pueden calcular).

$$\begin{aligned} a_3 &= a_2 + a_1 = 2 + 1 = 3 \\ a_4 &= a_3 + a_2 = 3 + 2 = 5 \\ a_5 &= a_4 + a_3 = 5 + 3 = 8, \end{aligned}$$

y así sucesivamente. Esto es un ejemplo de la definición de una sucesión en forma *recursiva*.

La relación de recurrencia

$$a_{n+1} = 3a_n, \ n \geq 0$$

no define una única sucesión numérica. La sucesión $5, 15, 45, 135, \cdots$ satisface la relación de recurrencia anterior. Y la sucesión $7, 21, 63, 189, \cdots$ también la satisface. Para distinguir la sucesión particular descrita por $a_{n+1} = 3a_n$, necesitamos conocer uno de los términos de la sucesión. Por tanto,

$$\begin{aligned} &a_0 = 5, \\ &a_{n+1} = 3a_n, \forall n \geq 0 \end{aligned}$$

define en forma única la sucesión $5, 15, 45, \cdots$, mientras que

$$\begin{aligned} &a_0 = 7, \\ &a_{n+1} = 3a_n, \forall n \geq 0 \end{aligned}$$

identifica a $7, 21, 63, \cdots$.

Nuestro último ejemplo muestra la importancia de la condición inicial para determinar una única solución.

En algunas ocasiones resulta útil disponer de una *fórmula explícita* para a_n. Por ejemplo, podría ser

$$a_n = 3n + 2 \quad \text{o} \quad a_n = (n+1)(n+2).$$

De hecho convenimos en decir que hemos resuelto la recurrencia cuando disponemos de tal fórmula.

(Una razón por la que una fórmula explícita puede ser útil es para describir el comportamiento de una sucesión para valores grandes de n.)

2.2. Problemas de Recurrencias

Los métodos recursivos son fundamentales en el análisis de algoritmos. Estos métodos surgen cuando queremos resolver un problema dado, al descomponerlo o referirlo a problemas similares más pequeños. En lenguaje de programación podemos implementarlos mediante el uso de funciones y procedimientos recursivos, que permiten llamarse a sí mismos. Dedicaremos la presente sección a resolver problemas de recurrencias.

2.2.1. Las Torres de Hanoi

Consideremos n discos circulares, con diferentes diámetros, y agujeros en su centro. Estos discos pueden apilarse en cualquiera de los postes que se muestran en la Figura 2.2.1. Los discos están organizados por tamaño, de modo que el más pequeño se encuentra en la parte superior, encima de los otros discos, y el más grande en la parte inferior, debajo de todos. El objetivo es pasar los discos, de uno en uno, de modo que la torre original termine en un poste diferente. Cada uno de los postes puede utilizarse para ubicar de forma temporal los discos, pero no se permite que un disco más grande quede encima de otro más pequeño, en ningún poste. ¿Cuál es el número mínimo de movimientos necesarios para hacer esto con n discos?

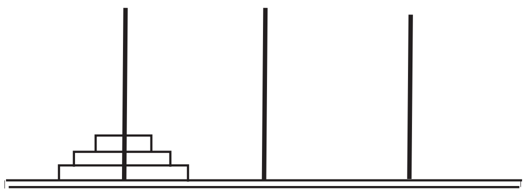

Figura 2.1: *Esta es la posición inicial en el problema de las torres de Hanoi para $n = 3$. Los tres discos están organizados en el poste de la izquierda. El objetivo es pasar los discos, de uno en uno, de modo que la torre original termine en un poste diferente.*

Veamos gráficamente (Figura 2.2.1), para $n = 3$ discos, una solución al problema que emplea 7 movimientos.

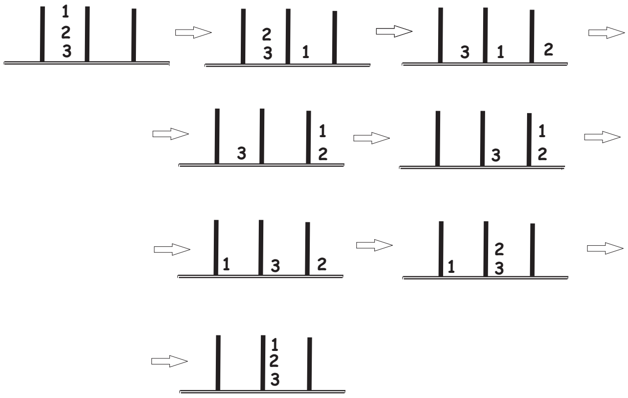

Figura 2.2: *Solución gráfica al problema de las torres de Hanoi con tres discos.*

Encontrar una relación de recurrencia

El problema de las Torres de Hanoi se puede resolver de modo recurrente como sigue.

Para $n \geq 0$, sea T_n el número mínimo de movimientos necesarios para mover una torre de n-discos, a un poste diferente. Entonces, para $n+1$ discos, haremos lo siguiente:

En algún momento debemos mover el disco más grande desde el primer poste a un poste diferente. Esto requiere 1 paso. Para que esto ocurra, los n discos más pequeños deben estar todos apilados en el poste restante. Esto se puede hacer en T_n pasos. Después de que el disco más grande se ha movido, se requieren otros T_n movimientos para organizar los restantes discos más pequeños encima.

Este argumento prueba que el número de pasos requeridos para completar el problema de las Torres de Hanoi con $n+1$ discos es:

$$T_{n+1} = 2T_n + 1$$

Sabiendo que $T_1 = 1$, podemos utilizar esta recurrencia para concluir que $T_2 = 3, T_3 = 7, T_4 = 15, \cdots$.

Resolver la recurrencia

El problema de las Torres de Hanoi fue planteado en 1883 por el matemático francés Edouard Lucas. En su planteamiento original habían 64 discos de oro.

Esta claro que calcular T_{64} utilizando la recurrencia implicaría mucho trabajo. Por lo tanto, sería interesante tener una expresión para T_n que nos permitiera obtener de forma rápida el número de pasos necesarios para resolver el problema de las Torres de Hanoi para cualquier número de discos.

A continuación, resolveremos el problema de las Torres de Hanoi utilizando dos métodos diferentes. El primer método que aplicaremos para resolver este problema consiste en conjeturar la solución y entonces verificar, generalmente con una demostración por inducción, que la conjetura es correcta. A este método le llamaremos de *sustitución*.

Método de sustitución

Conjeturar

Para ayudarnos a realizar una conjetura acertada, podemos tabular T_n para valores pequeños de n:

n	T_n
1	1
2	3
3	7
4	15
5	31

A partir de los datos presentados en la tabla, podemos conjeturar:

$$T_n = 2^n - 1$$

Verificar

Para comprobar que la conjetura que hemos realizado no es errónea, la demostraremos por inducción sobre n.

Sea $P(n)$ la propiedad que establece que $T_n = 2^n - 1$.

El primer caso, $P(1)$ es cierto porque $T_1 = 1 = 2^1 - 1$. Hagamos una hipótesis de inducción, y supongamos que la propiedad es cierta para un valor específico de n, digamos $n = k$, con lo que $T_k = 2^k - 1$. Y veamos cómo aplicando esta hipótesis podemos probar que el resultado también es correcto para $n = k + 1$.

$$\begin{aligned} T_{k+1} &= 2T_k + 1 \\ &= 2(2^k - 1) + 1 \\ &= 2^{k+1} - 1 \end{aligned}$$

Nuestra conjetura ahora está verificada. Esto prueba, por ejemplo, que el problema de las Torres de Hanoi con $n = 7$ discos requerirá $2^7 - 1 = 127$ movimientos para ser completado.

En general, el mejor método que tenemos para resolver recurrencias es el de sustitución. El único inconveniente del mismo, sería llegar a conjeturar la solución correcta. En el problema de las Torres de Hanoi, conjeturar la solución fue fácil, sin embargo algunas veces la solución tiene una expresión tan complicada que es bastante difícil poder llegar a conjeturarla. Presentaremos a continuación un método alternativo que nos permitirá resolver recurrencias. Este método se conoce como de *iteración* o *expansión*.

Método de Iteración

Consta de cuatro pasos, los cuales pasamos a detallar:

1. *Aplicar y simplificar*. Expandir la relación de recurrencia, sustituyendo el término con índice inferior por la expresión que dicta la relación de recurrencia y simplificando la expresión resultante. Hacer esto en varias ocasiones.

2. *Identificar y verificar el modelo*. Identificar un modelo para la ecuación de recurrencia después de i etapas del paso anterior. Y a continuación, verificar que este modelo es correcto tras expandir la ecuación de recurrencia en una etapa más.

3. *Expresar T_n en función de los primeros términos*. Sustituir un valor de i en el modelo de modo que T_n sea expresado como una función de los primeros términos.

4. *Encontrar una fórmula explícita para T_n*. Utilizando las condiciones iniciales, en la fórmula obtenida en el paso anterior, se resuelve la recurrencia.

Para ilustrar cómo trabaja el método de iteración, lo aplicaremos para resolver el problema de las Torres de Hanoi.

Paso 1: Aplicar y simplificar

$$\begin{array}{rcll} T_n & = & 1 + 2T_{n-1} & \\ & = & 1 + 2(1 + 2T_{n-2}) & \text{(aplicar)} \\ & = & 1 + 2 + 4T_{n-2} & \text{(simplificar)} \\ & = & 1 + 2 + 4(1 + 2T_{n-3}) & \text{(aplicar)} \\ & = & 1 + 2 + 4 + 8T_{n-3} & \text{(simplificar)} \\ & = & 1 + 2 + 4 + 8(1 + 2T_{n-4}) & \text{(aplicar)} \\ & = & 1 + 2 + 4 + 8 + 16T_{n-4} & \text{(simplificar)} \end{array}$$

La primera ecuación es la relación de recurrencia que queremos resolver. En el primer paso, transformamos la recurrencia original, aplicando la ecuación de recurrencia para hacer que T_n dependa de términos inferiores, eliminando T_{n-1} y haciendo que T_n dependa de T_{n-2}. A continuación simplificamos la expresión para T_n. Entonces aplicamos de nuevo la recurrencia, para que T_n dependa del término T_{n-3}. Simplificamos la expresión y volvemos a aplicar la recurrencia y simplificar. Paramos aquí de aplicar y simplificar porque parece quedar claro cuál será el modelo a seguir, tras sucesivas aplicaciones y simplificaciones.

Paso 2: Identificar y verificar el modelo

El modelo parece haber quedado claro: T_n es una suma de potencias consecutivas de 2, la última de ellas multiplicada por un término de la sucesión, de índice inferior a n, y relacionado con dicha potencia.

El próximo paso será describir este modelo, escribiendo la expresión tras i etapas de aplicar y simplificar la recurrencia. Y entonces realizaremos una última etapa (aplicar y simplificar) para confirmar que el modelo presentado es correcto

$$\begin{array}{ccl} T_n & = & 1 + 2 + 4 + \cdots + 2^{i-1} + 2^i T_{n-i} \\ \text{(aplicar)} & = & 1 + 2 + 4 + \cdots + 2^{i-1} + 2^i(1 + 2T_{n-(i+1)}) \\ \text{(simplificar)} & = & 1 + 2 + 4 + \cdots + 2^{i-1} + 2^i + 2^{i+1}T_{n-(i+1)} \end{array}$$

¡Ser cuidadosos a la hora de simplificar, dado que demasiada simplificación puede obscurecer el modelo emergente!

Paso 3: Expresar T_n en función de los primeros términos

La condición inicial en nuestro ejemplo es T_1. Si realizamos i pasos, se obtiene una expresión donde el término de índice menor es T_{n-i}. ¿Cuánto debe valer i para que $T_{n-i} = T_1$? La respuesta es $i = n - 1$.

$$\begin{array}{rcl} T_n & = & 1 + 2 + 4 + \cdots + 2^{i-1} + 2^i T_{n-i} \\ \text{(i=n-1)} & = & 1 + 2 + 4 + \cdots + 2^{n-2} + 2^{n-1} T_1 \end{array}$$

Esto nos lleva a una expresión de T_n no recurrente.

Paso 4: Encontrar una fórmula explícita para la expresión de T_n

Todos los pasos anteriores nos llevan a encontrar una forma cerrada para la expresión de T_n . Y en este caso, al ser $T_1 = 1$, hemos sido afortunados porque T_n es la suma de una serie geométrica.

$$\begin{array}{rcl} T_n & = & 1 + 2 + 4 + \cdots + 2^{n-2} + 2^{n-1} \\ & = & \displaystyle\sum_{i=0}^{n-1} 2^i \\ & = & 2^n - 1 \end{array}$$

2.2.2. Ordenación por el Método de la Burbuja

Uno de los métodos de ordenación más sencillos se conoce como la *ordenación de la burbuja*. Como veremos este problema se resolverá utilizando el método de sustitución.

El algoritmo

En este caso, la entrada es una lista $A[1], A[2], \cdots, A[n]$ de n números que deben ordenarse de forma ascendente. La idea es comparar términos adyacentes de la lista e intercambiarlos si están en el orden erróneo.

El segmento de programa que aparece en la Figura 2.3 proporciona una implementación para este tipo de algoritmo. En este caso, la variable entera i es el contador para el ciclo For exterior, mientras que la variable entera j es el contador para el ciclo For interior. Por último, la variable real *temp* se usa para guardar lo necesario para hacer un intercambio.

```
Begin
    For i:=1 to n-1 do
      For j:=n downto i+1 do
        If A[j] < A[j − 1] then
          Begin
            temp:=A[j − 1];
            A[j − 1] := A[j];
            A[j] := temp
          End
End;
```

Figura 2.3:

Comparamos el último elemento de la lista dada, $A[n]$, con su predecesor, $A[n-1]$. Si $A[n] < A[n-1]$, intercambiamos los valores guardados en $A[n-1]$ y $A[n]$. En todo caso, ahora tendremos que $A[n-1] \leq A[n]$. Después comparamos $A[n-1]$ con su predecesor inmediato, $A[n-2]$. Si $A[n-1] < A[n-2]$, los intercambiamos y continuamos el proceso. Después de $n-1$ comparaciones de este tipo, el número más pequeño de la lista está en $A[1]$. A continuación repetimos este proceso para los $n-1$ números restantes guardados en la lista más pequeña $A[2], A[3], \cdots, A[n]$. De esta manera, cada vez (contada por i) que se realiza este proceso, el número más pequeño de la sublista sube (como una *burbuja*) hasta el frente de esa sublista.

Por ejemplo, la tabla siguiente muestra el efecto de la primera pasada en el orden inicial 4, 7, 3, 1, 5 ,8, 2, 6.

Orden inicial	4	7	3	1	5	8	2	6
Después de la 1ª comparación	7	4						
Después de la 2ª comparación		4	3					
Después de la 3ª comparación			3	1				
Después de la 4ª comparación				5	1			
Después de la 5ª comparación					8	1		
Después de la 6ª comparación						2	1	
Después de la 7ª comparación							6	1
Después de la primera pasada	7	4	3	5	8	2	6	1

Encontrar una recurrencia

Para determinar la función de complejidad en tiempo cuando usamos este algoritmo en una lista de tamaño $n \geq 1$, contamos el total de comparaciones realizadas para ordenar los n números dados. Si a_n denota el número de comparaciones necesarias para ordenar n números de esta forma, entonces tendremos la siguiente relación de recurrencia:

$$\begin{aligned} &a_1 = 0 \\ &a_n = a_{n-1} + n - 1, \;\; n \geq 2 \end{aligned}$$

Esto surge de lo siguiente: dada una lista de n números, hacemos $n-1$ comparaciones para subir el número más pequeño al principio de la lista. La sublista restante de $n-1$ elementos requiere entonces a_{n-1} comparaciones para ordenarse completamente.

Resolver la recurrencia

Para resolver esta relación aplicaremos el método de sustitución.

Método de Sustitución

Conjeturar

Para ayudarnos a realizar una conjetura acertada enumeraremos algunos términos, para ver si hay un patrón que podamos reconocer.

$$\begin{aligned} a_1 &= 0 \\ a_2 &= a_1 + (2-1) = 1 \\ a_3 &= a_2 + (3-1) = 1 + 2 \\ a_4 &= a_3 + (4-1) = 1 + 2 + 3 \\ &\vdots \end{aligned}$$

A partir de los datos anteriores podemos conjeturar:

$$a_n = 1 + 2 + \cdots + (n-1) = \frac{(n-1)n}{2}$$

Verificar

Para comprobar que la conjetura que hemos realizado no es errónea, la demostraremos por inducción sobre n.

Sea $P(n)$ la propiedad que establece que $a_n = \frac{(n-1)n}{2}$. El primer caso, $P(1)$ es cierto porque $a_1 = 0 = \frac{(1-1)1}{2}$. Hagamos una hipótesis de inducción, y supongamos que la propiedad es cierta para un valor específico de n, digamos $n = k$, con lo que $a_k = \frac{(k-1)k}{2}$. Y veamos cómo aplicando esta hipótesis podemos probar que el resultado también es correcto para $n = k+1$.

$$\begin{aligned} a_{k+1} &= a_k + ((k+1) - 1) \\ &= \tfrac{(k-1)k}{2} + k \\ &= \tfrac{(k-1)k+2k}{2} \\ &= \tfrac{k(k+1)}{2} \end{aligned}$$

Lo cual prueba nuestra conjetura.

Notemos que este problema también podría resolverse utilizando el método de iteración. Se deja como ejercicio para el lector.

2.2.3. Algoritmo de Ordenación por Fusión

Hay muchos algoritmos para ordenar una lista de n artículos. Uno de los más populares es el algoritmo de ordenación por fusión (Merge sort).

El algoritmo

A continuación presentaremos cómo trabaja dicho algoritmo:

La entrada es una lista de $n \geq 1$ artículos $x_1, x_2, \cdots, x_n$.

- Si $n = 1$, entonces el algoritmo devuelve el artículo simple x_1.

- Si $n > 1$, entonces la lista original se divide en dos sublistas,

$$x_1, x_2, \cdots, x_{n/2} \quad \text{y} \quad x_{(n/2)+1}, \cdots, x_n.$$

  Ambas se ordenarán recursivamente, y entonces se mezclarán para formar una lista ordenada de los n artículos originales.

Veamos con un ejemplo cómo trabaja el algoritmo anterior.

Sublista 1	Sublista 2	Lista
5, 7, 10, 23	2, 3, 4, 9	
5, 7, 10, 23	3, 4, 9	2
5, 7, 10, 23	4, 9	2, 3
5, 7, 10, 23	9	2, 3, 4
7, 10, 23	9	2, 3, 4, 5
10, 23	9	2, 3, 4, 5, 7
10, 23		2, 3, 4, 5, 7, 9
		2, 3, 4, 5, 7, 9, 10, 23

Figura 2.4: *Contenidos de las listas en cada paso del algoritmo de ordenación por fusión*

Ejemplo 2.2

Supongamos que queremos ordenar la lista siguiente: 10, 7, 23, 5, 2, 4, 3, 9. Dado que el número de elementos es más grande que 1, dividiremos la lista en dos; una es 10, 7, 23, 5, y la otra 2, 4, 3, 9. Cada sublista se ordena recursivamente. Los resultados serán 5, 7, 10, 23 y 2, 3, 4, 9. Ahora debemos fusionar estas dos sublistas ordenadas en una lista ordenada mayor. Empezaremos con la lista vacia, y en cada paso añadiremos un número, resultado de comparar el primer elemento de las sublistas. De los dos números comparados, nos quedamos con el que sea más pequeño y lo añadimos al final de la lista grande. Este proceso se repite, hasta que una de las sublistas se queda sin elementos. En ese momento, la sublista restante se añade al final de la lista grande y por tanto queda acabada la tarea.

La Figura 2.4 muestra los contenidos de las listas en cada uno de los pasos del algoritmo.

Encontrar una recurrencia

Al analizar un algoritmo para ordenar números, una pregunta habitual es ¿cuál será el número máximo de comparaciones utilizadas al ordenarlos? El número de comparaciones se toma como una estimación del tiempo de ejecución -complejidad- del algoritmo (the run time). En el caso del algoritmo de ordenación por fusión, podemos encontrar una recurrencia para esta cantidad. Resolver esta recurrencia nos permitirá estudiar la conveniencia asintótica del algoritmo.

Para hacer el análisis más simple, supondremos que el número de elementos que estamos ordenando es una potencia de dos. Esto nos asegura que podemos dividir la lista exactamente por la mitad en cada etapa de la recursión.

Sea T_n el número máximo de comparaciones utilizadas por el algoritmo para ordenar una lista de n elementos. Si $n > 1$, entonces esto es igual al número de comparaciones utilizadas para ordenar ambas mitades de la lista, más el número de comparaciones utilizados para fusionar ambas mitades.

El número de comparaciones utilizados para ordenar cada mitad de la lista es $T_{n/2}$. Ordenar ambas mitades requiere por tanto $2T_{n/2}$ comparaciones. El número de comparaciones utilizadas para fusionar ambas mitades de la lista de n elementos es como mucho $n - 1$. La razón es que un número es añadido a la lista grande después de cada comparación, y al menos un número es añadido a la lista grande en el último paso, cuando una de las listas más pequeñas se ha quedado sin elementos. Desde luego, el número de comparaciones podría ser inferior, pero recordemos que estamos analizando el peor de los casos.

Añadiendo todas las comparaciones obtendremos la siguiente recurrencia:

$$T_n = 2T_{n/2} + n - 1$$

Ninguna comparación se requiere para ordenar una lista con un único elemento, por lo tanto $T_1 = 0$. A partir de este hecho y de la recurrencia, podremos calcular $T_2 = 2T_1 + 2 - 1 = 1, T_4 = 2T_2 + 4 - 1 = 5$, etc.

Resolver la recurrencia

Nos gustaría encontrar una fórmula explícita para el número de comparaciones utilizadas por el algoritmo de ordenación por fusión al ordenar una lista de n artículos. Esto requiere resolver la recurrencia $T_n = 2T_{n/2} + n - 1$. Para intentarlo utilizaremos el método de iteración presentado en la sección anterior, dado que a partir de los primeros valores de T_n (0, 1, 5, 17, 49) es difícil conjeturar un modelo y, por lo tanto, el método de sustitución no lo podremos aplicar.

<u>*Método de Iteración*</u>

Paso 1: aplicar y simplificar

$$\begin{array}{ccl}
T_n & = & n-1+2T_{n/2} \\
\text{(aplicar)} & = & n-1+2((n/2)-1+2T_{n/4}) \\
\text{(simplificar)} & = & n-1+n-2+4T_{n/4} \\
\text{(aplicar)} & = & n-1+n-2+4(n/4-1+2T_{n/8}) \\
\text{(simplificar)} & = & n-1+n-2+n-4+8T_{n/8} \\
\text{(aplicar)} & = & n-1+n-2+n-4+8(n/8-1+2T_{n/16}) \\
\text{(simplificar)} & = & n-1+n-2+n-4+n-8+16T_{n/16}
\end{array}$$

Al aplicar y simplificar la recurrencia en tres ocasiones, parece quedar claro cuál será el modelo a seguir.

$$T_n = n-2^0+n-2^1+n-2^2+n-2^3+n-2^4T_{n/2^4}$$

Paso 2: verificar el modelo

$$\begin{array}{ccl}
T_n & = & n-2^0+n-2^1+\cdots+n-2^{i-1}+2^iT_{n/2^i} \\
\text{(aplicar)} & = & n-1+n-2+\cdots+n-2^{i-1}+2^i(n/2^i-1+2T_{n/2^{i+1}}) \\
\text{(simplificar)} & = & n-1+n-2+\cdots+n-2^{i-1}+n-2^i+2^{i+1}T_{n/2^{i+1}}
\end{array}$$

Paso 3: expresar T_n en función de los primeros términos

Debemos sustituir un valor para i dentro del modelo, de modo que T_n sólo dependa de los términos más pequeños. Una elección natural sería $i = \log_2 n$, ya que $T_{n/2^i} = T_1$. Esta sustitución hace que T_n sólo dependa de T_1, el cual sabemos que vale 0.

$$\begin{array}{rcl}
T_n & = & n-2^0+n-2^1+\cdots+n-2^{i-1}+2^iT_{n/2^i} \\
 & = & n-1+n-2+\cdots+n-2^{\log_2(n)-1}+2^{\log_2 n}T_{n/2^{\log_2 n}} \\
 & = & n-1+n-2+\cdots+n-2^{\log_2(n)-1}+2^{\log_2 n}T_1 \\
 & = & n-1+n-2+\cdots+n-2^{\log_2(n)-1}
\end{array}$$

Paso 4: Encontrar una fórmula explícita para la expresión de T_n

Ahora tenemos una expresión no recurrente para T_n. Obtengamos su suma.

$$\begin{aligned}
T_n &= n-1+n-2+\cdots+n-2^{\log_2(n)-1} \\
&= \sum_{i=0}^{\log_2(n)-1} n-2^i \\
&= \sum_{i=0}^{\log_2(n)-1} n-\sum_{i=0}^{\log_2(n)-1} 2^i \\
&= n\log_2 n-(2^{\log_2 n}-1) \\
&= n\log_2 n-n+1 \\
&\sim n\log_2 n
\end{aligned}$$

Ya tenemos la solución. ¿La habríamos conjeturado? Comprobemos cómo con esta fórmula obtenemos los mismos valores para T_n que los que calculamos anteriormente.

n	T_n	$n\log_2 n-n+1$
1	0	$1\log_2 1-1+1=0$
2	1	$2\log_2 2-2+1=1$
4	5	$4\log_2 4-4+1=5$
8	17	$8\log_2 8-8+1=17$
16	49	$16\log_2 16-16+1=49$

Notemos que las $n\log_2 n$ comparaciones utilizadas por el algoritmo de ordenación por fusión están lejos del total de comparaciones posibles $\frac{n(n-1)}{2}$.

2.2.4. Técnica Divide y Vencerás

Comparemos las recurrencias del problema de las Torres de Hanoi y el algoritmo de ordenación por fusión.

$$\begin{array}{ll}
T_n = 2T_{n-1}+1 & \qquad T_n \sim 2^n \\
T_n = 2T_{n/2}+n-1 & \qquad T_n \sim n\log_2 n
\end{array}$$

En el problema de las Torres de Hanoi, dividimos un problema de tamaño n en dos subproblemas de tamaño $n-1$ y añadimos únicamente un paso adicional (mover el disco más grande). En el algoritmo de ordenación por fusión, dividimos un problema de tamaño n en dos subproblemas de tamaño $n/2$ y añadimos $n-1$ pasos adicionales (refundir dos sublistas).

¿Qué algoritmo es más rápido? El de ordenación por fusión. El punto clave es que generar subproblemas más pequeños es más importante que reducir los pasos adicionales por etapa recursiva.

La técnica utilizada en el algoritmo de ordenación por fusión se conoce con el nombre de *divide y vencerás*. Se trata de una de las técnicas algorítmicas básicas más potentes. Veamos otro ejemplo en el cual se aplica la misma.

Algoritmo de búsqueda binaria

Consideremos, por ejemplo, el algoritmo de búsqueda binaria, el cual describiremos en el contexto de acertar un número entre 1 y 100. Supongamos que alguien nos pide que acertemos un número entre 1 y 100, y después de cada intento fallido, nos dice si el número a acertar es superior o inferior al número que nosotros habíamos elegido.

El modo natural de proceder es empezar por 50. ¿Por qué? Porque si no acertamos, sabremos si el número se encuentra entre 51 y 100 o entre 1 y 50. En cualquier caso ahora tenemos que acertar un número en un rango de números que es la mitad de grande. Así hemos *dividido* el problema en otro problema que es la mitad de grande, y ahora podemos (recursivamente) *vencer* este problema.

Si denotamos por $T(n)$ al número de intentos realizados hasta acertar en búsqueda binaria un número sobre n números, y suponiendo que n es una potencia de 2, una recurrencia para $T(n)$ vendrá dada por

$$\begin{aligned} &T(1) = 1, \\ &T(n) = T(n/2) + 1 \text{ para } n \geq 2. \end{aligned}$$

Esto es, el número de intentos en búsqueda binaria hasta acertar un número de entre n es igual a: un paso para decidir inicialmente en qué mitad del conjunto se halla, más el tiempo que se invierte en hacer búsqueda binaria sobre $\frac{n}{2}$ números escogidos.

2.3. Recurrencias Lineales

En la presente sección, consideraremos una familia de recurrencias, las llamadas *recurrencias lineales*, las cuales aparecen de forma frecuente en problemas relacionados con la programación de ordenadores.

Definición 2.2

Una relación de recurrencia se llama *lineal* con coeficientes constantes si es de la forma

$$a_n \quad = \quad c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_d a_{n-d} + f(n)$$

donde $c_1, c_2, \cdots, c_d$ son constantes, $f(n)$ es una función arbitraria, y d es el orden de la recurrencia. (Notar que d se puede obtener como la diferencia entre el subíndice de orden mayor y el de orden menor.) Si $f(n) = 0$, la relación de recurrencia se dice que es *homogénea*, mientras que si la función $f(n) \neq 0$ la relación es *no homogénea* .

En primer lugar veremos cómo resolver una recurrencia lineal homogénea específica y generalizaremos nuestro método de trabajo para todas las recurrencias lineales homogéneas. Y a continuación presentaremos el método de los coeficientes indeterminados que nos permitirá resolver un tipo particular de recurrencias lineales no homogéneas.

2.3.1. Recurrencia de Fibonacci

Los números de Fibonacci $F_0, F_1, F_2, \cdots$ aparecen en muchas aplicaciones y se definen recursivamente de la forma siguiente:

$$F_0 = 1, F_1 = 1$$
$$F_{n+2} = F_{n+1} + F_n, \ n \geq 0$$

Notemos que la recurrencia de Fibonacci es de orden 2 y tiene por constantes $c_1 = 1$, y $c_2 = 1$.

Resolver esta recurrencia es fácil porque la relación es lineal. (Bien, fácil en el sentido de que puedes aprender la técnica para resolverla en una lección; descubrirla fue un trabajo de seis siglos). El método que aplicaremos para resolverla será el de sustitución.

Método de Sustitución

Conjeturar

Para una recurrencia lineal, una solución exponencial podría ser una buena conjetura.

$$a_n = c\alpha^n$$

En la expresión anterior c y α son parámetros introducidos para probar nuestra conjetura.

Verificar

Sustituyendo nuestra conjetura en la relación de recurrencia

$$a_n = a_{n-1} + a_{n-2}$$

da:

$$\begin{aligned} c\alpha^n &= c\alpha^{n-1} + c\alpha^{n-2} \\ \alpha^2 &= \alpha + 1 \\ \alpha^2 - \alpha - 1 &= 0 \\ \alpha &= \tfrac{1\pm\sqrt{5}}{2} \end{aligned}$$

Este cálculo sugiere que c puede ser cualquiera, pero α debe ser $\frac{1\pm\sqrt{5}}{2}$. Evidentemente, hay dos soluciones a la recurrencia:

$$a_n = c\left(\frac{1+\sqrt{5}}{2}\right) \quad \text{o} \quad a_n = c\left(\frac{1-\sqrt{5}}{2}\right)$$

Hay más soluciones de las sugeridas por nuestra demostración; de hecho, cualquier combinación lineal de las dos soluciones anteriores también es una solución. El siguiente teorema establece que esto es cierto, en general, para recurrencias lineales homogéneas.

Teorema 2.1 *Si $a_n = \alpha_1^n$ y $a_n = \alpha_2^n$ son dos soluciones de una relación de recurrencia lineal homogénea, entonces*

$$a_n = k_1\alpha_1^n + k_2\alpha_2^n$$

también es una solución, para cualesquiera constantes k_1 y k_2.

Dem. Toda relación de recurrencia lineal homogénea tiene la siguiente estructura:

$$a_n = \sum_{i=1}^{d} c_i a_{n-i}$$

Debemos probar que $a_n = k_1\alpha_1^n + k_2\alpha_2^n$ es una solución; es decir, debemos probar:

$$k_1\alpha_1^n + k_2\alpha_2^n = \sum_{i=1}^{d} c_i(k_1\alpha_1^{n-i} + k_2\alpha_2^{n-i})$$

Sustituyendo las dos soluciones $a_n = \alpha_1^n$ y $a_n = \alpha_2^n$ en la relación de recurrencia obtendremos las dos ecuaciones siguientes:

$$\begin{array}{l} \alpha_1^n = \sum_{i=1}^{d} c_i \alpha_1^{n-i} \\ \alpha_2^n = \sum_{i=1}^{d} c_i \alpha_2^{n-i} \end{array}$$

Multiplicando la primera ecuación por k_1, la segunda por k_2, y sumándolas obtendremos el resultado deseado. $\square$

El teorema anterior implica que para todos k_1 y k_2, la siguiente expresión es una solución a la relación de recurrencia de Fibonacci:

$$a_n = k_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + k_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$$

Nuestro siguiente objetivo será encontrar una solución que satisfaga las condiciones iniciales $a_0 = 0$ y $a_1 = 1$. Sustituyendo $n = 0$ y $n = 1$ en a_n,

$$\left\{ \begin{array}{lcl} a_0 & = & k_1 \left(\frac{1+\sqrt{5}}{2}\right)^0 + k_2 \left(\frac{1-\sqrt{5}}{2}\right)^0 = 0 \\ a_1 & = & k_1 \left(\frac{1+\sqrt{5}}{2}\right)^1 + k_2 \left(\frac{1-\sqrt{5}}{2}\right)^1 = 1 \end{array} \right.$$

tendremos dos ecuaciones con dos incógnitas:

$$\left\{ \begin{array}{rcl} k_1 + k_2 & = & 0 \\ k_1 \left(\frac{1+\sqrt{5}}{2}\right) + k_2 \left(\frac{1-\sqrt{5}}{2}\right) & = & 1. \end{array} \right.$$

Resolviendo el sistema obtendremos $k_1 = \frac{1}{\sqrt{5}}$ y $k_2 = \frac{-1}{\sqrt{5}}$. Ahora tenemos una solución completa a la recurrencia de Fibonacci:

$$a_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n$$

Esta solución parece errónea, por aparecer en la expresión el término $\sqrt{5}$, si tenemos en cuenta que todos los números de Fibonacci son enteros. Si sustituimos en la expresión $n = 0, 1, 2, \cdots$ comprobaremos que las raíces cuadradas se simplifican y se obtienen los números de Fibonacci.

2.3.2. Recurrencias Lineales Homogéneas

El método utilizado para resolver la recurrencia de Fibonacci se puede utilizar para resolver cualquier recurrencia lineal homogénea. Recordemos que una relación de recurrencia lineal homogénea tiene la siguiente expresión:

$$a_n = \sum_{i=1}^{d} c_i a_{n-i}$$

Sustituyendo la conjetura $a_n = \alpha^n$ dentro de la recurrencia obtendremos:

$$\alpha^n \quad = \quad c_1\alpha^{n-1} + c_2\alpha^{n-2} + \cdots + c_d\alpha^{n-d}$$

Simplificando y despejando todos los términos a un mismo miembro de la igualdad obtendremos la llamada *ecuación característica* de la recurrencia.

$$\alpha^d - c_1\alpha^{d-1} - c_2\alpha^{d-2} - \cdots - c_{d-1}\alpha - a_d = 0$$

Notemos que los coeficientes de la ecuación característica son los mismos que los coeficientes de la recurrencia. En general, la ecuación característica tendrá d raíces, donde d es el orden de la recurrencia. La solución dependerá de si las raíces son iguales o diferentes.

Raíces diferentes

Si la ecuación característica tiene d raíces todas diferentes $r_1, r_2, \cdots, r_d$, entonces la solución a la recurrencia tiene la expresión:

$$a_n = k_1 r_1^n + k_2 r_2^n + \cdots + k_d r_d^n$$

Notemos que las raíces de la ecuación característica no necesitan ser reales. Al final, las partes imaginarias se cancelarán, del mismo modo que se cancelaban las $\sqrt{5}$ en la solución de la recurrencia de Fibonacci.

Las constantes $k_1, k_2, \cdots, k_d$ dependen de las condiciones iniciales. Las condiciones iniciales generan un sistema de ecuaciones lineales en estas constantes. Por ejemplo, las condiciones iniciales

$$a_0 = b_0, a_1 = b_1, \cdots, a_{d-1} = b_{d-1},$$

dan lugar al sistema:

$$\begin{aligned} b_0 &= k_1 + k_2 + \cdots + k_d \\ b_1 &= k_1 r_1 + k_2 r_2 + \cdots + k_d r_d \\ &\vdots \\ b_{d-1} &= k_1 r_1^{d-1} + k_2 r_2^{d-1} + \cdots + k_d r_d^{d-1} \end{aligned}$$

El cual nos permitirá encontrar un conjunto de constantes, $k_0, k_1, \cdots, k_{d-1}$, soluciones de la recurrencia, las cuales satisfacen las condiciones iniciales.

Raíces múltiples

Si la ecuación característica de una recurrencia tiene una raíz múltiple, entonces la solución es más complicada. La herramienta principal es el siguiente teorema, el cual no demostraremos.

Teorema 2.2 *Si r es una raíz de la ecuación característica de una relación de recurrencia con multiplicidad m, entonces $r^n, nr^n, n^2r^n, \cdots, n^{m-1}r^n$ son soluciones a la recurrencia.*

Por ejemplo, si la ecuación característica de una recurrencia tiene:

raíz r_1 con multiplicidad 2,
raíz r_2 con multiplicidad 1,
y raíz r_3 con multiplicidad 3,

entonces la solución a la recurrencia tiene la forma:

$$a_n = k_1 r_1^n + k_2 n r_1^n + k_3 r_2^n + k_4 r_3^n + k_5 n r_3^n + k_6 n^2 r_3^n$$

Como antes, las constantes dependen de las condiciones iniciales. Cada condición inicial genera una ecuación lineal, y las constantes se calculan resolviendo el sistema de ecuaciones lineales resultante.

Para comprender este tipo de recurrencias, veamos un ejemplo.

Ejemplo 2.3

Supongamos que hay un tipo de plantas que nunca mueren, pero únicamente se reproducen durante el primer año de vida. ¿Cuánto crecerá la población de la planta?

Sea a_n el número de plantas en el año n. Como condiciones iniciales tenemos que $a_0 = 0$ y $a_1 = 1$. La población de plantas en el año n es igual a la población de plantas del año anterior más el número de plantas nuevas. La población del año anterior es a_{n-1}. Y el número de plantas nuevas este año es igual al número de plantas nuevas del último año, a saber: $a_{n-1} - a_{n-2}$. Ahora podemos construir una relación de recurrencia para la población de las plantas:

$$\begin{aligned} a_n &= a_{n-1} + (a_{n-1} - a_{n-2}) \\ &= 2a_{n-1} - a_{n-2} \end{aligned}$$

Y despejando, tendremos la ecuación

$$a_n - 2a_{n-1} + a_{n-2} = 0$$

La ecuación característica es $\alpha^2 - 2\alpha + 1 = 0$, la cual tiene la siguiente raíz $\alpha = 1$ con multiplicidad 2. Por lo tanto, la solución a la relación de recurrencia vendrá dada por:

$$\begin{aligned} a_n &= k_1(1)^n + k_2 n(1)^n \\ &= k_1 + k_2 \cdot n \end{aligned}$$

Las condiciones iniciales $a_0 = 0$ y $a_1 = 1$, generan el siguiente sistema de ecuaciones lineales:

$$\begin{aligned} k_1(1)^n + k_2 \cdot 0 &= 0 \\ k_1 + k_2 \cdot 1 &= 1 \end{aligned}$$

La solución al sistema lineal es $k_1 = 0$ y $k_2 = 1$. Por tanto, la solución a la recurrencia será:

$$\begin{aligned} a_n &= k_1(1)^n + k_2 n(1)^n \\ &= 0 + 1 \cdot n \\ &= n \end{aligned}$$

2.3.3. Recurrencias Lineales No Homogéneas

Estudiaremos ahora las recurrencias lineales no homogéneas con coeficientes constantes.

Definición 2.3

Una relación de recurrencia lineal *no homogénea* con coeficientes constantes tiene la forma:

$$a_n - c_1 a_{n-1} - \cdots - c_d a_{n-d} = f(n)$$

donde $c_1, c_2, \cdots, c_d$ son constantes, y $f(n) \neq 0$.

Aunque no existe un método general para resolver todas las recurrencias no homogéneas, existe una técnica útil cuando la función $f(n)$ tiene cierta forma. Este método se conoce como el *método de los coeficientes indeterminados* y se basa en la relación homogénea asociada que se obtiene al reemplazar $f(n)$ por cero.

Método de los coeficientes indeterminados

Consta de tres pasos, que pasamos a describir a continuación:

1. *Resolver la relación de recurrencia homogénea.* Reemplazar $f(n)$ por 0 y resolver la relación de recurrencia homogénea resultante (ignorar las condicones iniciales por ahora). A la solución de la recurrencia homogénea le llamaremos *solución general de la recurrencia homogénea* asociada, y la denotaremos por $a_n^{(H)}$.

2. *Encontrar una solución particular.* Sustituir $f(n)$ y encontrar una solución a la recurrencia (de nuevo ignorando las condicones iniciales). A esta solución le llamaremos *solución particular de la recurrencia no homogénea*, y la denotaremos por $a_n^{(P)}$.

3. *Encontrar una solución general.* Juntar las soluciones particular y homogénea para obtener la *solución general de la recurrencia no homogénea.*
$$a_n = a_n^{(H)} + a_n^{(P)}$$
Ahora utilizaremos las condicones iniciales para determinar los factores constantes por el método habitual de generar y resolver un sistema de ecuaciones lineales.

Ejemplo 2.4

Sea la relación de recurrencia
$$\begin{cases} a_1 = 1 \\ a_n = 4a_{n-1} + 3^n, \forall n \geq 2, \end{cases}$$
resolvámosla. Para ello aplicaremos el método de los coeficientes indeterminados.

Paso 1: Resolver la relación de recurrencia homogénea

La relación de recurrencia homogénea asociada es

$$a_n - 4a_{n-1} = 0.$$

La ecuación característica es $\alpha - 4 = 0$. La cual tiene una única raíz $\alpha = 4$. Por tanto, la solución general de la relación de recurrencia homogénea vendrá dada por $a_n^{(H)} = k4^n$.

Paso 2: Encontrar una solución particular

Debemos encontrar una solución simple a la recurrencia

$$a_n = 4a_{n-1} + 3^n.$$

Esta es la parte complicada. Para obtener $a_n^{(P)}$ usamos la forma de $f(n)$ para sugerir una forma de $a_n^{(P)}$. Conjeturemos que hay una solución de la forma $A3^n$. Sustituyendo esto en la relación de recurrencia tendremos:

$$\begin{array}{rcl} a_n & = & 4a_{n-1} + 3^n \\ A3^n & = & 4A3^{n-1} + 3^n \\ 3A & = & 4A + 3 \\ A & = & -3 \end{array}$$

Ahora tenemos una solución particular de la relación de recurrencia no homogénea, $a_n = -3 \cdot 3^n = -3^{n+1}$.

Paso 3: Encontrar una solución general

Ahora añadiremos a la solución homogénea la solución particular, para obtener la solución general de la recurrencia no homogénea:

$$f(n) = k4^n - 3^{n+1}$$

El valor de la constante k lo obtendremos a partir de la condición inicial $a_1 = 1$:

$$\begin{array}{rcl} 1 & = & k4^1 - 3^{1+1} \\ & = & 4A - 9 \\ k & = & \frac{5}{2} \end{array}$$

La solución a la recurrencia lineal no homogénea es

$$a_n = \frac{5}{2}4^n - 3^{n+1}, \ n \geq 1.$$

Ejemplo 2.5

Consideremos la siguiente recurrencia lineal no homogénea

$$\begin{cases} a_0 = 4, a_1 = 7, \\ a_n = 4a_{n-1} - 4a_{n-2} + n, \forall n \geq 3, \end{cases}$$

y obtengamos su solución.

Paso 1: Encontrar una solución general

La relación de recurrencia homogénea es

$$a_n - 4a_{n-1} + 4a_{n-2} = 0.$$

La ecuación característica asociada a la relación de recurrencia homogénea es $\alpha^2 - 4\alpha + 4$, la cual tiene por solución $\alpha = 2$ con multiplicidad 2. Por lo tanto, la solución homogénea es $a_n^{(H)} = k_1 2^n + k_2 n 2^n$.

Paso 2: Encontrar una solución particular

Ahora necesitamos una solución particular de la relación de recurrencia no homogénea

$$a_n = 4a_{n-1} - 4a_{n-2} + n.$$

Conjeturamos una solución de la forma $A_1 n + A_0$ y la sustituimos en la recurrencia:

$$\begin{array}{rcl} a_n & = & 4a_{n-1} - 4a_{n-2} + n \\ A_1 n + A_0 & = & 4(A_1(n-1) + A_0) - 4(A_1(n-2) + A_0) + n \\ & = & 4A_1 n - 4A_1 + 4A_0 - 4A_1 n + 8A_1 - 4A_0 + n \\ & = & 4A_1 + n \end{array}$$

De donde obtenemos la solución particular $a_n^{(P)} = n + 4$.

Paso 3: Juntar las soluciones y encontrar las constantes

Juntando la solución general de la relación de recurrencia homogénea y la solución particular de la no homogénea obtendremos la solución general de la recurrencia no homogénea:

$$a_n = k_1 2^n + k_2 n 2^n + n + 4$$

Y a partir de las condiciones iniciales, $a_0 = 4$ y $a_1 = 7$, obtendremos los valores de las constantes.

$$\begin{array}{rcl} a_0 & = & k_1 2^0 + k_2 \cdot 0 \cdot 2^0 + 0 + 4 = 4 \\ a_1 & = & k_1 2^1 + k_2 \cdot 1 \cdot 2^1 + 1 + 4 = 7 \end{array}$$

De donde $k_1 = 0$ y $k_2 = 1$. En conclusión, la solución general a la recurrencia no homogénea será

$$a_n = n2^n + n + 4,\ n \geq 0$$

La parte más complicada al resolver recurrencias no homogéneas es encontrar una solución particular $a_n^{(P)}$. A continuación presentaremos unas ideas que nos facilitarán la obtención de la misma.

Solución particular de la recurrencia no homogénea

Supongamos que la recurrencia tiene la forma:

$$a_n - c_1 a_{n-1} - c_2 a_{n-2} - \cdots - c_d a_{n-d} = f(n)$$

Un método que funciona en la mayoría de los casos es conjeturar una solución particular con la misma forma que $f(n)$. Esto es, si $f(n)$ es un polinomio de grado t, entonces conjeturar que $a_n^{(P)}$ es un polinomio de grado t. Por ejemplo, si $f(n) = n^2$, entonces conjeturar $a_n^{(P)} = A_2 n^2 + A_1 n + A_0$ y determinar las constantes A_2, A_1 y A_0 sustituyendo y verificando. Análogamente si $f(n)$ es exponencial, entonces conjeturar que $a_n^{(P)}$ es exponencial.

La Tabla 2.5 resume la relación entre $f(n)$ y $a_n^{(P)}$. ¿Cómo utilizaremos esta tabla?

1. Si $f(n)$ es un múltiplo constante de una de las formas de la primera columna de la tabla 2.5 y no es solución de la relación homogénea asociada a la recurrencia, entonces $a_n^{(P)}$ toma la correspondiente expresión que se muestra en la segunda columna de la tabla 2.5.

$f(n)$	$a_n^{(P)}$
k, una constante	A, una constante
n	$A_1n + A_0$
n^2	$A_2n^2 + A_1n + A_0$
n^t, $t \in Z^+$	$A_tn^t + A_{t-1}n^{t-1} + \cdots + A_1n + A_0$
r^n, $r \in R$	Ar^n
n^tr^n	$r^n(A_tn^t + A_{t-1}n^{t-1} + \cdots + A_1n + A_0)$

Figura 2.5: *Tabla que proporciona conjeturas a las soluciones particulares $a_n^{(P)}$ en función de $f(n)$*

2. Cuando $f(n)$ es una suma de múltiplos constantes de términos como los de la primera columna de la tabla, y ninguno de éstos es una solución de la relación homogénea asociada, entonces $a_n^{(P)}$ se forma como la suma de los términos correspondientes en la columna encabezada por $a_n^{(P)}$. Por ejemplo, si $f(n) = n^2 + 3^n$ y ninguno de los sumandos de $f(n)$ es solución de la relación homogénea asociada, entonces

$$a_n^{(P)} = (A_2n^2 + A_1n + A_0) + A3^n.$$

3. Las cosas se complican si un sumando $f_1(n)$ de $f(n)$ es un múltiplo constante de una solución de la relación homogénea asociada. Si esto ocurre, multiplicamos la solución particular $a_{n_1}^{(P)}$ correspondiente a $f_1(n)$, por la mínima potencia de n, digamos n^s, para la que ningún sumando de $n^sf(n)$ sea una solución de la relación homogénea asociada. Entonces $n^sa_{n_1}^{(P)}$ es la parte correspondiente de la solución particular $a_n^{(P)}$. Ilustremos con un ejemplo la explicación anterior.

Ejemplo 2.6

Consideremos la relación de recurrencia lineal no homogénea

$$a_n + 4a_{n-1} + 4a_{n-2} = f(n), \ n \geq 2.$$

La componente homogénea de la solución es

$$a_n^{(H)} = k_1(-2)^n + k_2n(-2)^n$$

donde k_1 y k_2 son constantes arbitrarias. En consecuencia, *a*) si $f(n) = 5(-2)^n$, entonces $a_n^{(P)} = An^2(-2)^n$;

b) si $f(n) = 7n(-2)^n$, entonces $a_n^{(P)} = Bn^3(-2)^n + Cn^2(-2)^n$; pero

c) si $f(n) = -11n^2(-2)^n$, entonces $a_n^{(P)} = Dn^4(-2)^n En^3(-2)^n + Fn^2(-2)^n$;

Las constantes A, B, C se determinarían al sustituir $a_n^{(P)}$ en la relación de recurrencia no homogénea dada.

Bibliografía

1. [*Biggs*] Norman L. Biggs, *Matemática Discreta*, Ed. Vicens Vives, 1994.

2. [*Bogart*] Kenneth P. Bogart, *Matemáticas Discretas*, Ed. Limusa, 1996.

3. [*Garcia*] García Merayo F., *Matemática Discreta*, 2ªEdición Ed. Thomson, 2006.

4. [*Gimbert*] Gimbert J. y otros, *Apropament a la Teoria de Grafs i als seus Algorismes*, Edicions de la Universitat de Lleida, 1998.

5. [*Grassmann*] W. K. Grassmann y J. P. Tremblay, *Matemática Discreta y Lógica*, Ed. Prentice Hall, 1996.

6. [*Grimaldi*] Ralph P. Grimaldi, *Matemáticas Discreta y Combinatoria*, Ed. Addison-Wesley Iberoamericana, 1997.

7. [*Rosen*] Rosen K. H., *Discrete Mathematics and its Applications*, Ed. MacGraw-Hill International, 1999.

Problemas: Combinatoria

¡Un buen consejo! Cuando se trata de un problema de conteo, debemos preguntarnos acerca de la importancia del orden en el problema. Cuando el orden es necesario, pensamos en términos de permutaciones y en la Regla del Producto. Cuando el orden no sea necesario, las combinaciones podrían tener un papel importante en la solución del problema.

1. En la memoria principal de un ordenador, la información se almacena en celdas de memoria. Para identificar las celdas de memoria principal del ordenador, cada celda tiene asignado un nombre único conocido como su *dirección*. En algunos ordenadores, una dirección se representa mediante una lista ordenada de ocho símbolos, en la que cada símbolo es uno de los *bits* (de binary digits, dígitos binarios) 0 o 1. Esta lista de ocho bits se denomina *byte* ¿Cuántas direcciones tenemos por cada celda de memoria?

   (Solución pág. 5 [*Grimaldi*])

2. Se tienen tres tareas y cuatro ordenadores. Hay que asignar cada tarea a un solo ordenador, y ningún ordenador debe recibir más de una tarea. ¿De cuántas maneras se puede hacer esto?

   (Solución pág. 295 [*Grassmann*])

3. Al elegir un ordenador de cierta compañía, se tiene la opción de elegir cinco unidades diferentes de procesamiento central y tres pantallas diferentes. Estas dos componentes constituyen un sistema completo. ¿De cuántas formas se puede elegir el sistema?

   (Solución: 15 formas diferentes)

4. Matías trabaja como técnico de sistemas en una pequeña universidad. Una tarde, él ve que durante el día se han enviado 12 programas para

su procesamiento por lotes. ¿De cuántas formas puede ordenar Matías el procesamiento de estos programas si:

a) no existen restricciones?

b) él considera que cuatro de los programas tienen prioridad sobre los otros ocho y desea procesarlos antes?

c) primero separa los programas en los cuatro de máxima prioridad, cinco de menor prioridad y tres de mínima prioridad, y desea procesar los doce programas de modo que los de máxima prioridad se procesen primero y los tres programas de mínima prioridad se procesen al final?

(Solución: (a) 12!, (b) 4!8!, (c) 4!5!3!)

5. En una implementación del lenguaje de programación Pascal, un identificador consta de una sola letra, o de una sola letra seguida de hasta siete símbolos, que pueden ser letras o números. (Supongamos que el ordenador no distingue entre las letras mayúsculas y minúsculas; hay 26 letras y 10 números). Sin embargo, ciertas palabras clave están reservadas para los comandos; en consecuencia, estas palabras clave no pueden usarse como identificadores. Si esta implementación tiene 36 palabras reservadas, ¿cuántos identificadores diferentes son posibles en esta versión de Pascal?

   (Solución: $26 + 26(36) + 26(36)^2 + \cdots + 26(36)^7 - 36$)

6. La producción de una pieza de una máquina consta de cuatro etapas. Hay seis líneas de ensamble disponibles para la primera etapa, cuatro líneas para la segunda etapa, cinco para la tercera y cinco para la última. Determine la cantidad de formas diferentes en que dicha pieza puede quedar totalmente ensamblada en este proceso de producción.

   (Solución: 600 formas diferentes)

7. *a*) Determine el valor de la variable entera *counter* después de la ejecución del siguiente segmento de programa en Pascal. (Aquí *i*, *j* y *k* son variables enteras).

   counter :=0;

   For i:=1 to 12 do

   counter:=counter+1;

For j:=5 to 10 do

counter:=counter+2;

For k:=15 downto 8 do

counter:=counter+3;

b) ¿Qué principio de conteo está en juego en el apartado anterior?

(Solución: (a) *counter* = 48, (b) Principio de la Suma)

8. *a*) Considere el siguiente segmento de programa en Pascal, donde *i*, *j* y *k* son variables enteras

For i:= 1 to 12 do

For j:=5 to 10 do

For k:=15 downto 8 do

*Writeln ((i-j)*k);*

¿Cuántas veces se ejecuta la proposición *Writeln*?

b) ¿Qué principio de conteo se usó en el apartado anterior?

(Solución: (a) 576, (b) Principio del Producto)

9. Demuestre que si se colocan las 26 letras del alfabeto (cinco de las cuales son vocales) en círculo, hay al menos cinco consonantes consecutivas.

(Solución Aplicar el principio de las casillas: las 21 consonantes han de estar repartidas entre las 5 vocales $21 = 5 \cdot 4 + 1 > 5 \cdot 4$ luego debe haber un par de vocales entre las cuales haya al menos 5 consonantes)

10. En un conjunto de 100 naturales, ¿cuántos podemos asegurar que tendrán el mismo resto al dividir por 21?

(Solución Aplicar el principio de las casillas $100 = 21 \cdot 4 + 16 > 21 \cdot 4$ luego uno de los 21 restos posibles $(0, 1, \cdots, 20)$ debe ser el resto de al menos 5 de los 100 naturales)

11. Hallar los valores de $\binom{n}{0}, \binom{n}{1}, \binom{n}{n}$, para todo $n \geq 1$.

(Solución: (a) 1, (b) n, (c) 1)

12. Demostrar que $\binom{n}{m} = \binom{n}{n-m}$, para $0 \leq m \leq n$.

(Solución pág. 26 [*Grimaldi*])

13. ¿Cuál es el valor de $\binom{5}{3}$? Escriba todos los subconjuntos de tres elementos de un conjunto de 5 elementos para mostrar que hay $\binom{5}{3}$ de ellos.

    (Solución: 10)

14. Calcule lo siguiente: (a) $\binom{n}{n-2}$, (b) $\binom{n}{2}$, (c) $\binom{20}{2}$.

    (Solución: (a) $\frac{n(n-1)}{2}$, (b) $\frac{n(n-1)}{2}$, (c) 190)

15. Desarrolla $(x+y)^7$, calculando los coeficientes.

    (Solución: $(x+y)^7 = \displaystyle\sum_{i=0}^{7} \binom{7}{i} x^i y^{7-i}$, ahora podemos utilizar el triángulo de Tartaglia para calcular los coeficientes y obtenemos el desarrollo: $= y^7 + 7xy^6 + 21x^2y^5 + 35x^3y^4 + 35x^4y^3 + 21x^5y^2 + 7x^6y + x^7$)

16. En los estudios de la teoría algebraica de códigos y la teoría de lenguajes de computación, consideramos ciertas disposiciones, llamadas cadenas (*strings*), formadas a partir de un alfabeto prescrito de símbolos. Si el alfabeto ya determinado consta de los símbolos 0, 1 y 2, por ejemplo, entonces 01, 11, 21, 12 y 20 son cinco de las nueve cadenas de longitud dos. Entre las 27 cadenas de longitud tres están 000, 012, 202 y 110.

    En general, si n es un entero positivo, por la Regla del Producto, existen 3^n cadenas de longitud n para el alfabeto 0, 1 y 2. Si $x = x_1x_2\cdots x_n$ es una de esas cadenas, definimos el *peso* de x, que se denota $wt(x)$, como $wt(x) = x_1 + x_2 + \cdots + x_n$. Por ejemplo, si $n = 2$, $wt(12) = 3$ y $wt(22) = 4$; y si $n = 3$, $wt(101) = 2$, y $wt(222) = 6$.

    Entre las 3^{10} cadenas de longitud 10, determinar el número de cadenas que tengan peso par.

    (Solución pág. 23 [*Grimaldi*])

17. Calcule cuántas ternas (i, j, k) de números naturales cumplen

    $1 \le i < j < k \le 10$.

    (Solución Son combinaciones de 10 elementos tomados de 3 en 3, $C_{10}^3 = 120$, pues una vez escogidos los tres valores, sólo hay una manera de asignarlos a i, j y k)

18. Calcule cuántas ternas (i, j, k) de números naturales cumplen

    $1 \leq i \leq j \leq k \leq 10$.

    (Solución Son combinaciones con repetición de 10 elementos tomados de 3 en 3, $CR_{10}^{3} = 220$, pues una vez escogidos los tres valores, que pueden ser repetidos o no, sólo hay una manera de asignarlos a i, j y k)

19. Considere el siguiente segmento de un programa de Pascal, donde $i_1, i_2, \cdots, i_r$ son variables enteras

    For $i_1 := 1$ *to 30 do*

    For $i_2 := 1$ *to* i_1 *do*

    $\vdots$

    For $i_r := 1$ *to* i_{r-1} *do*

    *Writeln ((i-j)*k);*

    ¿Cuántas veces se ejecuta la proposición *Writeln* en este segmento de programa?

    (Solución pág. 39 [*Grimaldi*])

20. Consideremos el siguiente segmento de programa en Pascal, donde las variables i, j, n, y *counter* son enteras. Hemos supuesto que, en una sección anterior del programa, el usuario proporcionó un entero positivo para el valor de n.

    counter:=0

    For $i := 1$ *to* n *do*

    For $j := 1$ *to* i *do*

    counter:=counter+1;

    Después de que se ejecute este segmento de programa, ¿qué valor tendrá la variable *counter*?

    (Solución pág. 39 [*Grimaldi*])

21. Determine el número de trayectorias (escalonadas) del plano xy de $(2, 1)$ a $(7, 4)$. Notemos que cada trayectoria está formada por escalones individuales que van una unidad hacia la derecha (R) ó una unidad hacia arriba (U).

(Solución pág. 10 [*Grimaldi*])

22. Determine cuántos códigos pueden escribirse con 5 ceros, 3 unos y 2 espacios en blanco.

    (Solución hay tantos códigos como permutaciones de tres elementos con repeticiones fijadas: $P_3^{5,3,2} = 2520$)

23. Una caja de discos de un ordenador personal contiene una gran cantidad de discos de cuatro marcas diferentes. ¿De cuántas formas es posible elegir seis discos de la caja?

    (Solución pág. 299 [*Bogart*])

24. Determine el número de soluciones enteras no negativas de

$$x_1 + x_2 + x_3 + x_4 = 32,$$

    donde

    a) $x_i \geq 0,\ 1 \leq i \leq 4.$
    b) $x_i > 0,\ 1 \leq i \leq 4.$
    c) $x_1, x_2 \geq 5,\ x_3, x_4 \geq 7.$
    d) $x_i \geq -2,\ 1 \leq i \leq 4.$

    (Solución (a) $\binom{35}{32}$, (b) $\binom{31}{28}$, (c) $\binom{11}{8}$, (d) $\binom{43}{40}$)

25. Determine el número de soluciones enteras no negativas de la ecuación $x_1 + x_2 + \cdots + x_6 = 10$.

    (Solución pág. 36 [*Grimaldi*])

26. Determine el número de términos que aparecen en el desarrollo de $(x + y + z)^5$.

    (Solución $CR_3^5 = 21$)

27. Determine el coeficiente del término de xy^2z^2 en el desarrollo de $(x + y + z)^5$.

    (Solución $P_5^{1,2,2} = 30$)

28. ¿Cuántas permutaciones de $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ hay con la primera cifra impar y la última menor que 5?

    (Solución $3 \cdot 5P_8 + 2 \cdot 4P_8 = 23 \cdot 8!$)

29. ¿Cuántas permutaciones de $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ hay con el 5 no en primer lugar y el 9 no en el último?

    (Solución $P_{10} - |A_5 \cup A_9| = 73 \cdot 8!$, donde A_5 son las permutaciones que empiezan con 5 y A_9 son las que acaban con 9)

30. ¿Cuántas permutaciones de $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ hay con el 5 no en primer lugar o el 9 no en el último?

    (Solución Se trata del conjunto complementario de las permutaciones que empiezan con 5 y acaban con 9, esto es, $10\,! - P_8 = 89 \cdot 8!$)

Problemas: Recurrencia

1. Resuelva las siguientes recurrencias

   a) $a_0 = 1, a_1 = 3, a_n = 5a_{n-1} + 6a_{n-2}, n \geq 2$
   (Solución $a_n = \frac{3}{7}(-1)^n + \frac{4}{7}6^n$, $n \geq 0$)

   b) $a_0 = 2, a_1 = -8, 2a_{n+2} - 11a_{n+1} + 5a_n = 0, n \geq 0$
   (Solución $a_n = (-2)5^n + 4(\frac{1}{2})^n$, $n \geq 0$)

   c) $a_0 = 7, a_1 = 3, 3a_{n+1} = 2a_n + a_{n-1}, n \geq 1$
   (Solución $a_n = 4 + 3(-\frac{1}{3})^n$, $n \geq 0$)

   d) $a_0 = 0, a_1 = 3, a_{n+2} - a_n = 0, n \geq 0$
   (Solución $a_n = \frac{3}{2} + -\frac{3}{2}(-1)^n$, $n \geq 0$)

   e) $a_0 = 0, a_1 = 3, a_{n+2} + a_n = 0, n \geq 0$
   (Solución $a_n = 3 \sin \frac{n\pi}{2}$, $n \geq 0$)

   f) $a_0 = 1, a_1 = 3, a_{n+2} = 4a_{n+1} - 4a_n, n \geq 0$
   (Solución $a_n = 2^n + n2^{n-1}$, $n \geq 0$)

   g) $a_0 = 0, a_1 = 1, a_n = 5a_{n-1} - 6a_{n-2}, n \geq 2$
   (Solución $a_n = 3^n - 2^n$, $n \geq 0$)

   h) $a_0 = 1, a_1 = 1, a_{n+2} - 3a_{n+1} - 4a_n = 0, n \geq 0$
   (Solución $a_n = \frac{2}{5}4^n + \frac{3}{5}(-1)^n$, $n \geq 0$)

   i) $a_0 = -2, a_1 = 1, a_{n+2} - 2a_{n+1} + a_n = 0, n \geq 0$
   (Solución $a_n = -2 + 3n$, $n \geq 0$)

2. La solución de una recurrencia lineal con coeficientes constantes homogenea es: $a_n = 2^{n+1} - n3^n$. Busca la equación de recurrencia y las condiciones iniciales.

   (Solución $a_n = 8a_{n-1} - 21a_{n-2} + 18a_{n-3}$, $a_1 = 1$, $a_2 = -10$ y $a_3 = -65$)

3. Busca y resuelve una relación de recurrencia para el número de permutaciones de un conjunto de n elementos.

   (Solución rel. de rec. $a_n = na_{n-1}$, $n \geq 2 \quad a_1 = 1$, solución: $a_n = n!$)

4. Sea S un conjunto de $n = 2^k$ enteros distintos ($k \geq 1$). Demostrar que el máximo y el mínimo de S pueden hallarse con $f_n = \frac{3}{2}n - 2$ comparaciones binarias. (Solución pág. 284 [*Biggs*])

5. Sea q_n el número de palabras de longitud n en el alfabeto $\{0, 1\}$ con la propiedad de que no hay dos ceros consecutivos. demostrar que:

$$q_1 = 2, q_2 = 3,$$
$$q_n = q_{n-1} + q_{n-2}, n \geq 3$$

   (Solución pág. 283 [*Biggs*], pág. 474 [*Grimaldi*])

6. ¿Cuántos subconjuntos tiene el conjunto vacío? ¿Cuántos subconjuntos tiene el conjunto con un único elemento $\{1\}$? ¿Cuántos subconjuntos tiene el conjunto de cardinal 2 $\{1, 2\}$? ¿Cuántos de estos contienen el elemento 2? ¿Cuántos subconjuntos tiene el conjunto $\{1, 2, 3\}$? ¿Cuántos contienen el elemento 3? Obtener una recurrencia para el número de subconjuntos de un conjunto de cardinal n y probar por inducción que vuestra recurrencia es correcta. (Solución pág. 352 [*Bogart*])

7. Obtener una recurrencia para el número de biyecciones (funciones inyectivas y sobreyectivas) de $1, 2, \cdots, n$ a un conjunto de n elementos y probar por inducción que vuestra recurrencia es correcta. (Solución pág. 353 [*Bogart*])

8. En muchos lenguajes de programación podemos considerar que las expresiones aritméticas válidas, sin paréntesis, están formadas por los dígitos, $0, 1, 2, \cdots, 9$ y los símbolos de las operaciones binarias $+, *$, y $/$. Por ejemplo, $2+3/5$ es una expresión aritmética válida y $8+*9$ no lo es. Encontrar una recurrencia para el número de expresiones aritméticas válidas formadas por n símbolos. (Solución pág. 472 [*Grimaldi*])

9. Resuelva las siguientes recurrencias

   a) $a_0 = 2$, $a_n - 3a_{n-1} = 5(7^n), n \geq 1$
   (Solución $a_n = -\frac{1}{4}3^{n+3} + \frac{5}{4}7^{n+1}$, $n \geq 0$)

b) $a_2 = 1,\, a_{n+1} = a_n + n, n \geq 1$
(Solución $a_n = \frac{1}{2}n(n-1),\ n \geq 0$)

c) $a_0 = 1, a_1 = 1,\, a_{n+2} = a_n + n, n \geq 0$
(Solución $a_n = \frac{9}{8} + \frac{1}{8}(-1)^{n+1} + \frac{1}{4}n^2 - \frac{1}{2}n,\ n \geq 0$)

d) $a_0 = 1,\, a_{n+1} - a_n = 2n + 3, n \geq 0$
(Solución $a_n = 1 + n^2 + 2n,\ n \geq 0$)

e) $a_0 = 1,\, a_{n+1} - 2a_n = 2^n, n \geq 0$
(Solución $a_n = 2^n(1 + n),\ n \geq 0$)

f) $a_0 = 2, a_1 = 5,\, a_{n+2} - 10a_{n+1} + 21a_n = 6(3^n), n \geq 0$
(Solución $a_n = -\frac{1}{4}7^n + \frac{9}{4}3^n,\ n \geq 0$)

10. Resuelva las siguientes recurrencias

a) $a_1 = 1,\, a_n = 3a_{n-1} + n,\ n \geq 2$
(Solución $a_n = (3/4)3^n - (3/4) - (n/2),\ n \geq 1$)

b) $a_0 = 1,\, a_n = 2a_{n-1} + 3^n,\ n \geq 2$
(Solución $a_n = -2^{n+1} + 3^{n+1},\ n \geq 0$)

c) $a_0 = 1,\, a_1 = 5,\, a_n = 4a_{n-1} - 4a_{n-2} + 1,\ n \geq 3$
(Solución $a_n = n2^{n+1} + 1,\ n \geq 0$)

11. Resolver el problema de las Torres de Hanoi utilizando el método de los coeficientes indeterminados.

(Solución rel. rec. $a_n = 2a_{n-1}+1,\ n \geq 2,\quad a_1 = 1$ solución: $a_n = 2^n-1$)

12. Resolver la recurrencia obtenida en el algoritmo de la burbuja utilizando el método de los coeficientes indeterminados.

(Solución rel. rec. $a_n = a_{n-1} + n - 1,\ n \geq 2,\quad a_1 = 0$ solución: $a_n = -(1/2)n + (1/2)n^2$)

13. Un sistema de señales permite emitir ceros, unos y espacios en blanco. Los ceros van de dos en dos y los unos también. Encontrar la recurrencia que calcula el número de señales que se pueden emitir de longitud n (no olvidar las condiciones iniciales).

(Solución rel. rec. $a_n = 2a_{n-2} + a_{n-1},\ n \geq 3,\quad a_1 = 1,\ a_2 = 3$)