Estructura de árbol y computación: tratamiento computacional del problema de los cuatro colores

Antonio MORENO REAL *

Es indudable que la utilización de diagramas de flujo para representar algoritmos nos ayuda a reconocer su estructura interna. Más aún, la intención que se preste a la estructura de un algoritmo tiene generalmente como consecuencia una mejor comprensión del proceso de computación, y con frecuencia produce el reconocimiento de alternativas y perfeccionamientos de la idea original.

De hecho, existe una conexión palpable entre los pasos necesarios para expresar un algoritmo y el camino elegido para razonar acerca del problema planteado. Del mismo modo, la habilidad para pensar sobre la estructura de un conjunto de datos y considerar cómo sus representaciones alternas pueden afectar la naturaleza de los algoritmos que tratan con dichos datos, facilitará la labor y mejorará los resultados.

Algunos tipos estructurales de datos, comúnmente utilizados, son los vectores y las matrices. Otro tipo de estructura, válida también para procesos de un algoritmo, son los denominados «árboles» (figura nº 1) cuyas principales características estructurales son:

1. Cada árbol posee un nudo inicial o raíz del que parten y se extienden segmentos a otros nudos, que se ramifican a su vez, llegando finalmente a un nudo terminal u hoja.

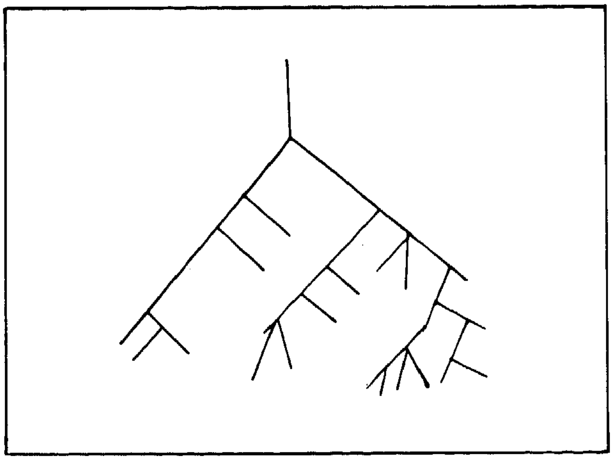


Fig. 1.—Árbol invertido.

2. Se puede considerar una «rama» como un subárbol, aunque este concepto es más general, que tiene por tanto su propio nudo inicial. En adelante supondremos que las estructuras de árbol son dibujadas desde la raíz hacia abajo. En este supuesto, diremos que una rama es todo lo que cuelga desde un nudo no terminal u hoja.

* Profesor agregado de bachillerato del I.C. «Carlos III», de Águilas (Murcia).

3. Los segmentos de cada rama se conectan siempre a nudos nuevos, que forman una continuación de la misma rama.
4. No existe un ciclo que regrese a los nudos más cercanos a la raíz.
5. No existe cruzamiento o entrecruzamiento entre las ramas.

Las estructuras de árbol son importantes en la representación de ciertos tipos de datos y, aunque parezca extraño, los pasos de un gran número de problemas matemáticos y juegos exhiben una estructura de árbol en sus algoritmos correspondientes.

Búsqueda en un árbol

Los árboles se recorren, ya sea para obtener información específica, para llegar a una conclusión, o para modificar el árbol en cierta forma. La búsqueda de un árbol es la idea principal de un gran número de problemas matemáticos y de juegos.

Se podrían estudiar varios procedimientos de búsqueda en un árbol, pero entre todos ellos existe uno que sistemáticamente recorre todos los nudos de un árbol, útil para buscar una solución en aquellos problemas que requieran un enfoque sistemático; lo llamaremos búsqueda en el «orden natural» y sigue las siguientes reglas:

1. Empiece en la raíz y no se detenga hasta llegar a una hoja o nudo terminal.
2. Una vez llegado a una hoja, regrese al nudo por el que acaba de pasar (nudo padre).
3. Avance nuevamente hacia delante, por cualquier segmento no considerado, hasta llegar a otro nudo terminal.
4. Si no quedan segmentos por considerar, regrese al nudo predecesor (padre) y repita el proceso para llegar a otra hoja.
5. Si resulta que usted está nuevamente en la

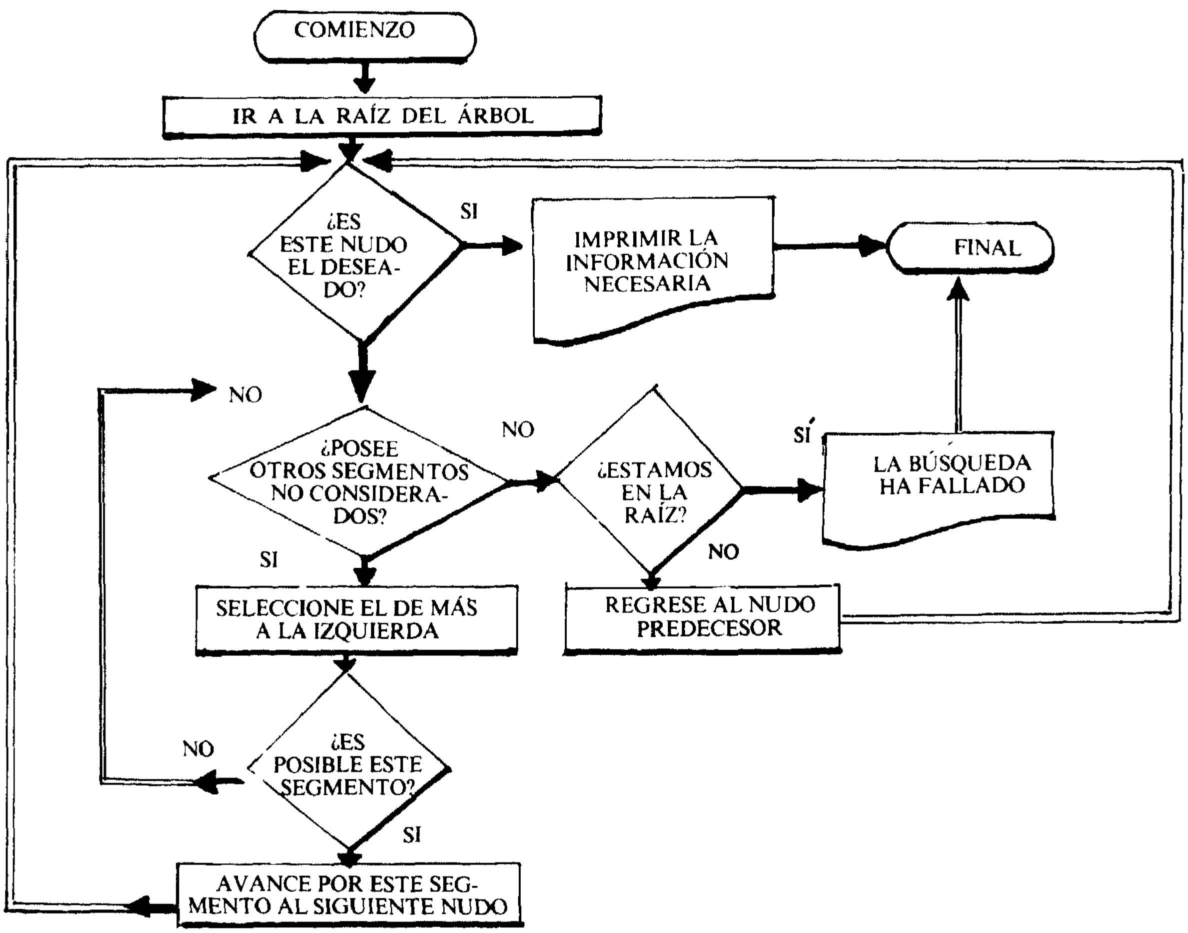


Fig. 2.—Diagrama de flujo para la búsqueda natural en árboles.

raíz y ya ha recorrido todos sus segmentos, la búsqueda ha terminado y el árbol completo ha sido recorrido en el orden natural.

Evidentemente a la hora de aplicar la búsqueda natural en un árbol como procedimiento para resolver un problema, éste debe admitir un algoritmo cuyo proceso de ejecución tenga una estructura de árbol, pero además, para que sea eficaz la búsqueda, tenemos que saber cuándo una rama o segmento es inadmisible para llegar a la solución deseada; hecho que se refleja en el diagrama de flujo para la búsqueda natural de la figura nº 2. De otro modo, sería necesario el recorrer todo el árbol para encontrar una solución al problema planteado.

A continuación estudiaremos un problema interesante que utiliza este tipo de búsqueda en su solución algorítmica.

La conjetura de los cuatro colores

Los mapas se colorean para facilitar el ver de forma inmediata la extensión de cada país. Claramente, resulta necesario que los países vecinos reciban colores diferentes. ¿Se puede colorear cualquier mapa con sólo cuatro colores?

Este problema es uno de los retos más celebrados en matemáticas. Sobre todo, desde que en 1978 el matemático Arthur Cayley, incapaz de probar o refutar la conjetura de los cuatro colores, presentó el problema en la London Mathematical Society; desde entonces, muchos matemáticos han intentado encontrar una demostración que afirmase la conjetura o un mapa mínimo (en cuanto al nº de países) que necesitase 5 colores para su coloreado, lo cual negaría la conjetura. Es fácil comprobar que la conjetura para tres colores es falsa, ya que es sencillo encontrar un mapa que requiera para su coloreado un mínimo de cuatro colores (ver fig. nº 3). Existe una demostración

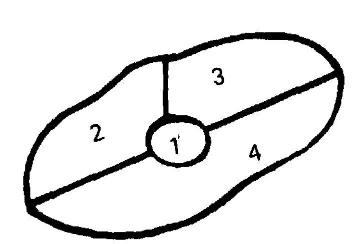


Figura 3.

muy sencilla en el libro «What is Mathematics? por Courant y Robbins de que la conjetura para cinco colores es cierta; sin embargo, no existe ninguna demostración breve, utilizando procedimientos convencionales, de que la conjetura de los cuatro colores sea cierta. Sí existe una demostración por Kenneth Appel y Wolfgang Hken en 1976 (*Scientific American*, de 1977), en la cual utilizando unos cálculos extremadamente extensos, por lo que se hace imprescindible el uso del ordenador y además algunas ideas cruciales en la demostración fueron perfeccionadas mediante experimentos computarizados, logran probar la conjetura de los 4 colores.

Después de esta introducción al problema de los cuatro colores, pretendemos aplicar el cálculo computacional a este problema construyendo un algoritmo y su programa correspondiente para que, una vez introducidos los datos que describen el mapa, en el microordenador, éste nos responda si el mapa se puede colorear con cuatro colores y de qué modo.

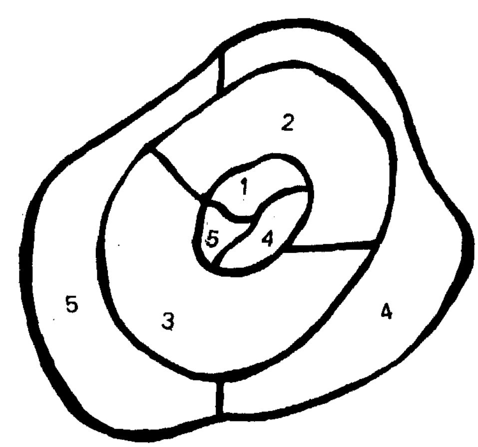


Figura 3 bis.

El problema de los cuatro colores como búsqueda en un árbol

Emplearemos como ilustración un ejemplo de mapa que verifique unas condiciones exigibles a un posible mapa mínimo de cinco colores. Por supuesto, el mapa debe ser «normal» en el sentido de que todo país debe ser una región plana conexa; de otro modo es fácil dibujar un mapa mínimo de cinco colores, (ver fig. nº 3 bis). Estas condiciones son: A) Ningún punto es frontera de más de 3 países. B) Cada país es vecino de cuando menos otros cinco. El mapa ejemplo se ilustra en la figura nº 4.

Podemos modelar el problema de colorear con cuatro colores el mapa de la figura nº 4, como el de búsqueda en un árbol (ver fig. nº 5) en el cual, cada segmento representa una decisión para colorear un país, cuyo color elegido se codifica con un

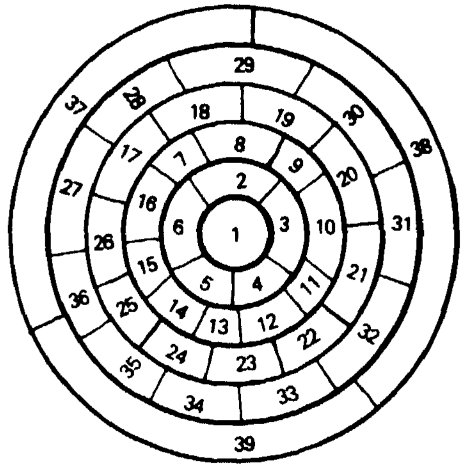
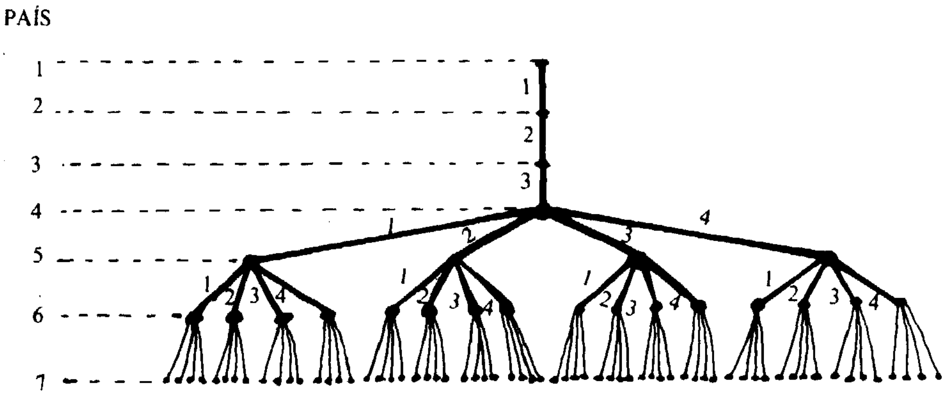


Figura 4.

nº entre el 1 y el 4; el i-ésimo segmento en una trayectoria desde la raíz corresponde al coloreado del i-ésimo país del mapa, por tanto todos los nudos del árbol situados al mismo nivel, medido desde la raíz, representan al mismo país. Podemos elegir siempre tres países vecinos entre sí, y empezar coloreándolos; como cada uno de éstos debe ser entonces de color diferente y no tiene importancia qué color le asignemos, escogeremos los colores 1, 2 y 3 tal como muestra el árbol de la fig. nº 5



(Figura n.º 5. Árbol correspondiente al coloreo de mapa de figura n.º 4)

Al colorear los países restantes es conveniente imaginarnos que para cada uno existen cuatro selecciones posibles; sin embargo, la mayoría de las veces, sólo serán admisibles, una, dos o tres de estas selecciones.

Preparación del algoritmo

El primer paso en la elaboración del algoritmo es la numeración de los países. Para una mayor eficacia, procuraremos que cada país tenga fronteras con aquel inmediatamente anterior en la numeración, o cuando menos con otro que tenga un nº inferior, tal como muestra el mapa de la fig. nº 4, y requeriremos que los tres primeros países sean vecinos entre sí.

El primer problema que se nos plantea es la introducción de datos, o sea, ¿cómo representar el mapa en la memoria del microordenador? Un procedimiento es mediante una matriz de conexiones, de modo que la fila i-ésima de la matriz enliste, en orden creciente, la numeración de los países vecinos del país nº i. El algoritmo que pretendemos construir consultará esta matriz de conexiones a la hora de colorear un país; más concretamente, en el mapa de la fig. nº 4, en el momento de tener que asignarle un color al país nº 20, el algoritmo tendrá que consultar la fila nº 20 de la matriz de conexiones, que tendrá esta estructura: (9, 10, 19, 21, 30, 31), y entonces, en función de los colores que hayamos asignado a estos países, el algoritmo deberá asignar el color adecuado para el país nº 20. No obstante, es lógico pensar que a la hora de colorear el país nº 20 basta con conocer los colores asignados a los países vecinos cuya numeración es inferior a 20, ya que los restantes países vecinos no han sido aún coloreados. Entonces, mejorando nuestra matriz de conexiones, bastará con guardar en la fila i-ésima los números que representan a los países vecinos del país nº i cuya numeración sea menor que i; el resto de la fila la completaremos con ceros.

La dimensión de la matriz de conexiones será pues de n × m donde n representa el número de países y m el nº máximo de vecinos de un país cualquiera del mapa.

Además necesitaremos un vector color (cr) de dimensión n que nos sirva para ir almacenando el código del color definitivo asignado a cada país.

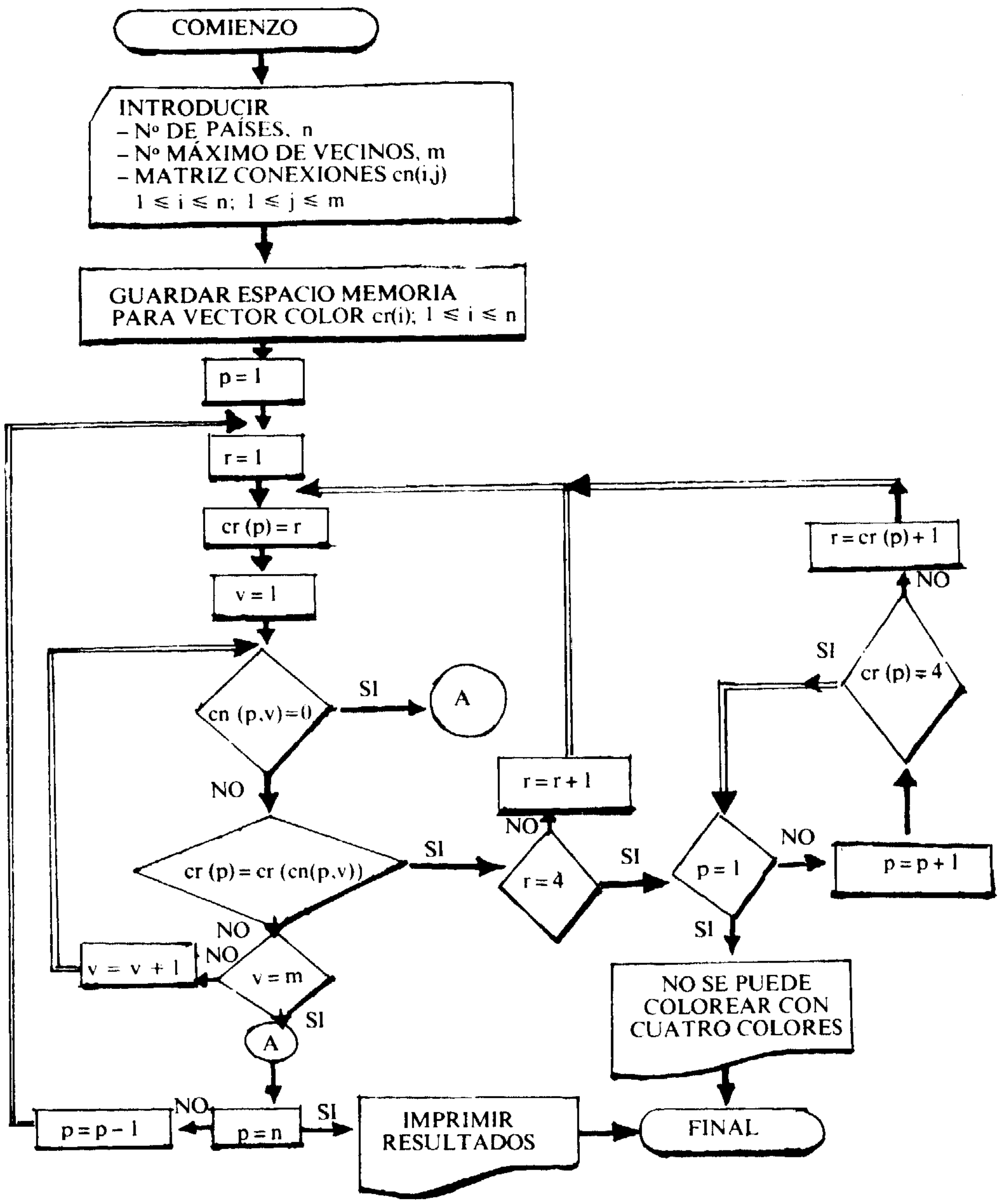
Observemos que una vez formada la matriz de conexiones ya podemos introducir todos los datos necesarios (el mapa) en el microordenador.

Ahora, teniendo en cuenta el diagrama de flujo de búsqueda natural en un árbol (fig. nº 2) y la estructura de árbol de nuestro problema (fig. nº 5), ya estamos en condiciones de desarrollar el algoritmo cuyo diagrama de flujo (fig. nº 6) deberá seguir los siguientes pasos:

1.º) Introducción de datos: nº de países (n), nº máximo de vecinos (m) y matriz de conexiones cn (i, j) de orden n × m.

2.º Colorear el país nº 1, dentro del ciclo que se desarrolla en el apartado 3º

3.º) Para colorear el país i-ésimo seguiremos el siguiente proceso: a) Le asignamos el color nº 1. b) Comparamos el color asignado con el de sus veci-



Fugura nº 6: Diagrama de flujo correspondiente al algoritmo que nos resuelve el problema de averiguar si un mapa se puede colorear con cuatro colores, y de qué manera, en caso afirmativo por supuesto.

nos ya coloreados. c) Si no coincide con ninguno de ellos, pasamos a colorear el siguiente país. d) Si el color asignado coincide con el de algún país vecino, entonces le asignamos el color siguiente y volvemos al apartado b).

4.º En el caso de que, cuando estemos coloreando el país nº i, no sea válido ninguno de los cuatro colores, preguntaremos si este país es el nº 1 (o sea, si estamos en la raíz):

a) En caso afirmativo, la búsqueda ha fallado y, por tanto, el mapa no se puede colorear con cuatro colores.

b) En caso negativo, volveremos al nudo predecesor (país anterior) y preguntaremos si este país ha sido coloreado con el último color (el nº 4); en caso negativo le asignaríamos el color siguiente y volveríamos al apartado 3º b), y en caso afirmativo volveríamos al apartado 4º.

5.º) Una vez coloreado el país nº n, el algoritmo ha terminado y podemos imprimir el vector color, obteniendo una solución, que no tiene por qué ser única, del problema.

De este modo, el diagrama de flujo para el coloreado de un mapa con cuatro colores sería tal y como lo muestra la figura nº 6.

A continuación del diagrama, listamos el programa correspondiente en lenguaje B.A.S.I.C., y mostramos los resultados obtenidos en un microordenador (New Brain 32K de RAM) al introducirle este programa con los datos del ejemplo descrito en la fig. nº 4, con lo que finalizamos este trabajo.

```
10 REM «COLOREAR UN MAPA CON
   CUATRO COLORES»
20 INPUT («CUÁNTOS PAÍSES?») n
30 INPUT («MÁXIMO VECINOS?») m
40 OPTION BASE 1: DIM cn (n, m): DIM cr
   (n)
50 FOR i = 1 TO n
60 FOR j = 1 TO m
70 PRINT «cn («;i;», «;j;») = »;
80 INPUT cn (i, j)
90 NEXT j
100 NEXT i
110 p = 1
120 r = 1
130 cr (p) = r
140 v = 1
150 IF cn (p, v) = 0 THEN 200
160 IF cr (p) = cr (cn(p, v.)) THEN 310
170 IF v = m THEN 200
180 v = v + 1
190 GO TO 150
200 IF p = n THEN 230
210 p = p + 1
220 GO TO 120
230 PRINT «SE PUEDE COLOREAR CON»;
240 PRINT «CUATRO COLORES»
250 PRINT,
260 PRINT «PAÍS»; TAB (19); «COLOR»
270 FOR p = 1 TO n
280 PRINT TAB (2); p; TAB (13); cr (p)
290 NEXT p
300 GO TO 500
310 IF r = 4 THEN 340
320 r = r + 1
330 GO TO 130
340 IF p = 1 THEN 390
350 p = p - 1
360 IF cr (p) = 4 THEN 340
370 r = cr (p) + 1
380 GO TO 130
390 PRINT «NO SE PUEDE COLOREAR
    CON»;
400 PRINT «CUATRO COLORES»
500 END
```

Resultados obtenidos al introducir en el microordenador el programa anterior con los datos del ejemplo mostrado en la figura nº 4

PAÍS:	COLOR	PAÍS:	COLOR
1	1	21	3
2	2	22	4
3	3	23	2
4	2	24	4
5	3	25	3
6	4	26	4
7	1	27	3
8	3	28	1
9	1	29	3
10	2	30	1
11	1	31	2
12	3	32	1
13	1	33	3
14	2	34	1
15	1	35	2
16	3	36	1
17	2	37	2
18	4	38	3
19	2	39	4
20	4		

Bibliografía

CLARK, K y COWEL, D.: *Ordenadores, programas y procesos de cálculo.* Editorial Pirámide. Madrid, 1976.

FORSYTHE, KEENAN, ORGANK y STENBERG.: *Lenguajes de diagramas de flujo.* Editorial Limusa. México, 1980.

SCHRIBES, THOMAS J.: *Fundamentos de diagramas de flujo.* Editorial Limusa. México, 1980.