La Torre d'Hanoi. Aplicació d'un procediment recursiu per a la solució d'un problema combinatori

JAUME RIBERA
Dept. d'Estadística
Facultat d'Informàtica de Barcelona

En aquesta nota exposem el concepte de recursivitat per a buscar solucions a alguns problemes combinatoris, en contraposició a la recerca exhaustiva, estudiant a fons l'exemple de la Torre d'Hanoi.

DESCRIPCIÓ DEL PROBLEMA

El problema de la Torre d'Hanoi es pot descriure així:

Tenim 3 clavilles, A, B, C, i n discs de n diferents diàmetres. Tots els discs tenen un forat al centre de tal forma que es poden apilar a qualsevol de les clavilles. Inicialment tots els discs són col·locats a la clavilla A en ordre de mida, el més gros a baix i el més xic a dalt.

El problema consisteix a transferir tots els discs de la pila A a la B emprant el clau C com a lloc de pas intermedi. Com a restriccions del nostre problema, cal dir que a cada moviment només es pot moure un disc d'un clau a un altre i que mai no pot haver-hi un disc de diàmetre superior damunt d'un altre de diàmetre inferior.

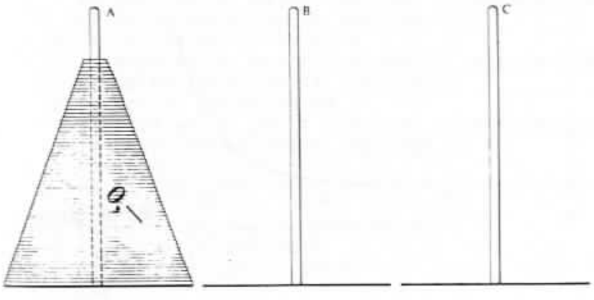


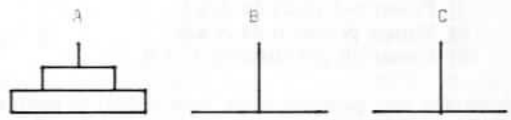
FIG 1

El nom d'aquest problema ve de la llegenda que conta que prop d'Hanoi hi ha uns monjos que treballen per trobar la solució amb 64 discs i que el dia que ho aconsegueixin s'acabarà el món. Si això fos veritat no ens hauríem d'amoïnar pas. Encara que poguessin fer moviments perfectes, sense errades, un per segon, tardarien milions i milions d'anys per aconseguir-ho.

Nosaltres però, simplificarem el problema i en donarem una resolució quan n = 2, 3, i un procediment general recursiu que ens permetrà de trobar la solució per a qualsevol n, demostrant al mateix temps inductivament que la solució existeix.

ARBRE D'ESTATS

Per al nostre cas quan n = 2, el problema consisteix a passar d'aquesta configuració



a aquesta:

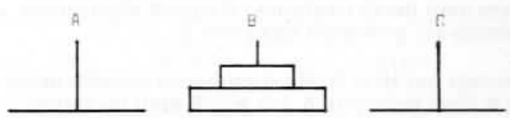


FIG 2

Per a representar els moviments possibles que ens aniran descrivint els estats, emprarem un arbre, on els nusos representaran els estats i els arcs els moviments.

Així, per a n = 2, la fig. 3 ens mostra l'arbre d'exploració fins a un nivell de profunditat de 3 moviments.

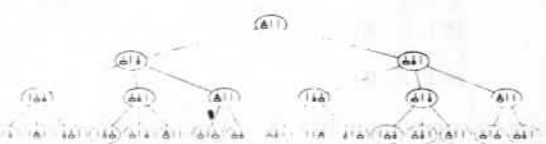


FIG 3

En aquest arbre podem observar que malgrat que hi figurin 25 nusos d'estat, només n'hi ha 9 de diferents. Com que l'estat representa totalment la configuració, i com que els estats idèntics representen punts de coneixement idèntic, podrem plegar el nostre arbre en aquells nusos on l'estat coincideix. Això ens du a representar la recerca mitjançant un graf.

És el que mostrem a la fig. 4.

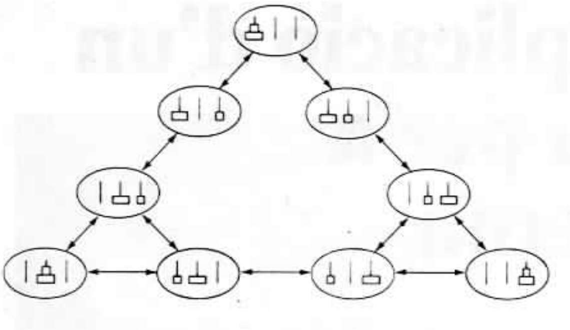


FIG 4

El nombre d'estats possibles creix desmesuradament en augmentar n.

Haurem de cercar un altre mètode per a la resolució del nostre problema.

SOLUCIÓ RECURSIVA

Si el nostre problema inicial amb n discs el subdividim en 3 passos, se'ns obrirà una nova visió del problema.

Si tenim n discs a A i els volem passar a B, això equival a

i) Passar n-1 discs de A a C.
ii) Moure el disc n de A a B.
iii) Passar els n-1 discs de C a B.

Amb aquest nou punt de vista, hem reduït el problema de la Torre d'Hanoi de n discs a dues vegades el mateix problema amb n-1 discs.

Si apliquem recursivament aquesta descomposició arribarem a la simplificació final de resoldre el problema de la Torre d'Hanoi quan n = 1, que és trivial i només cal moure el disc d'una clavilla a l'altra.

Veiem amb detall l'aplicació d'aquest algorisme a la resolució del problema quan n = 3.

Denotem per H(n, A, B) el problema consistent en moure n discs de la pila A a la pila B amb les restriccions de la Torre d'Hanoi, i per B → A el moviment del disc superior de la pila B a la pila A.

Així, el nostre problema és H(3, A, B), que equival a

H(2, A, C)
A → B
H(2, C, B)

Un pas posterior de la descomposició ens dóna:

H(1, A, B)
A → C
H(1, B, C)
A → B
H(1, C, A)
C → B
H(1, A, B)

Si tenim en compte que H(1, X, Y) equival a X → Y, la solució al problema de la Torre d'Hanoi H(3, A, B) és:

A → B
A → C
B → C
A → B
C → A
C → B
A → B

Aquests moviments corresponen a la fig. 5.

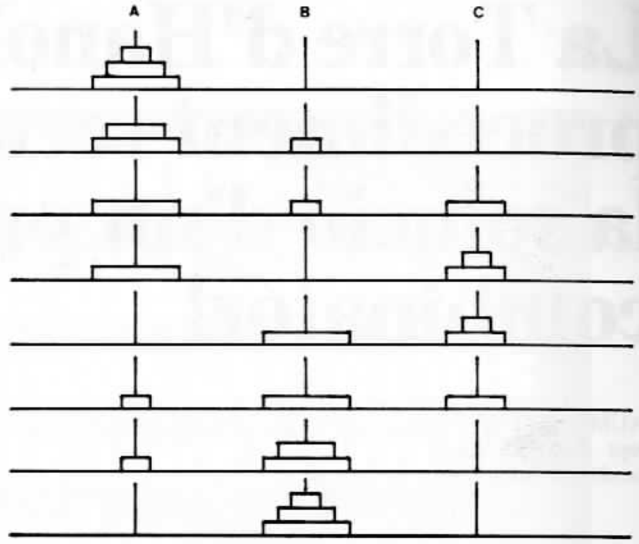


FIG 5

L'annex 1 ens mostra un llistat dels programes FORTRAN-V recursiu per a la resolució del problema de la Torre d'Hanoi per a qualsevol n, i un exemple d'execució quan n = 3, 4, 5.

Fàcilment es demostra que el nombre de moviments per resoldre el problema amb n discs és $2^n - 1$, i tal com ja havíem avançat, per n = 64 l'ordre del nombre de moviments és 10^{19}, i si ho considerem a un interval d'un segon per moviment, ens resulta un temps total de l'ordre de $5.8494\ 10^{11}$ anys.

Els temps d'execució dels programes de l'annex 1 són:

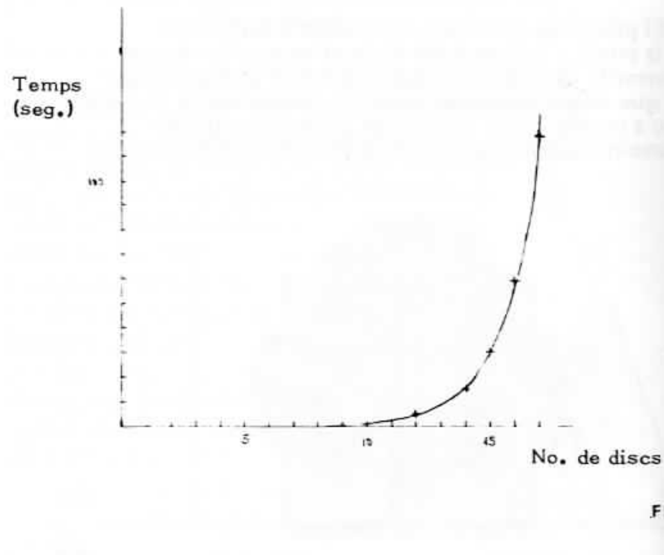


FIG 6

Anexe 1 LLISTAT DE PROGRAMES

```
C
C       RESOLUCIO DE LA TORRE D'HANOI
C
C       A,B,C SON LES PILES DE LES TRES CLAVILLES
C       A,B,C(0) ES EL NOMBRE D'ELEMENTS A CADA PILA
C       A,B,C(-1) ES EL NUMERO DE PILA
C
        DIMENSION A(-1:20),B(-1:20),C(-1:20)
        ACCEPT "NUMERO DE DISCS=",NUM
        A(0)=NUM
        B(0)=0
        C(0)=0
        A(-1)=1
        B(-1)=2
        C(-1)=3
        DO 1 I=1,NUM
      1 A(I)=NUM+1-I
        CALL TEMPS(10)
        CALL MOSTRA (A,B,C)
        CALL HANOI (NUM,A,B,C)
        CALL TEMPS(10)
        CALL EXIT
        END

        SUBROUTINE HANOI (N,A,B,C)
C
C       ENS RESOL LA TORRE D'HANOI PER N DISCS
C
        DIMENSION A(-1:20),B(-1:20),C(-1:20)
        IF (N.EQ.1) GO TO 1
        N1=N-1
        CALL HANOI(N1,A,C,B)
        CALL MOURE (A,B,C)
        CALL HANOI (N1,C,B,A)
        RETURN
      1 CALL MOURE (A,B,C)
        RETURN
        END

        SUBROUTINE MOURE (A,B,C)
C
C       MOU UN DISC DE LA PILA A A LA PILA B
C
        DIMENSION A(-1:20),B(-1:20),C(-1:20)
        J=A(0)
        J1=B(0)+1
        B(J1)=A(J)
        A(0)=A(0)-1
        B(0)=J1
        CALL MOSTRA (A,B,C)
        RETURN
        END

        SUBROUTINE MOSTRA (A,B,C)
C
C       IMPRIMEIX L'ESTAT DE LES TRES CLAVILLES
C
        DIMENSION A(-1:20),B(-1:20),C(-1:20)
        DO 1 J=1,3
        IF (A(-1).NE.J) GO TO 3
        IF(A(0).NE.0) WRITE (12,100) J,(A(K),K=1,A(0))
        IF (A(0).EQ.0) WRITE (12,130) J
      3 IF (B(-1).NE.J) GO TO 2
        IF(B(0).NE.0) WRITE (12,100) J,(B(K),K=1,B(0))
        IF (B(0).EQ.0) WRITE (12,130) J
      2 IF (C(-1).NE.J) GO TO 1
        IF(C(0).NE.0) WRITE (12,100) J,(C(K),K=1,C(0))
        IF (C(0).EQ.0) WRITE (12,130) J
      1 CONTINUE
        WRITE (12,120)
        RETURN
    100 FORMAT (/1X,1H*,I1,2H*I,20I3)
    120 FORMAT(/,1X,60(1H-))
    130 FORMAT(/1X,1H*,I1,2H*I)
        END
```

1.er cas, n = 3

```
*1*I  3  2  1      *1*I  3            *1*I  1

*2*I               *2*I               *2*I  3  2

*3*I               *3*I  2  1         *3*I

----------------   ----------------   ----------------

*1*I  3  2         *1*I               *1*I

*2*I  1            *2*I  3            *2*I  3  2  1

*3*I               *3*I  2  1         *3*I

----------------   ----------------   ----------------

*1*I  3            *1*I  1

*2*I  1            *2*I  3

*3*I  2            *3*I  2

----------------   ----------------
```

2.ª cas, n = 4

```
*1*I  4  3  2  1   *1*I  4  1         *1*I  2

*2*I               *2*I  2            *2*I  4  1

*3*I               *3*I  3            *3*I  3

----------------   ----------------   ----------------

*1*I  4  3  2      *1*I  4  1         *1*I  2  1

*2*I               *2*I               *2*I  4

*3*I  1            *3*I  3  2         *3*I  3

----------------   ----------------   ----------------

*1*I  4  3         *1*I  4            *1*I  2  1

*2*I  2            *2*I               *2*I  4  3

*3*I  1            *3*I  3  2  1      *3*I

----------------   ----------------

*1*I  4  3         *1*I               *1*I  2

*2*I  2  1         *2*I  4            *2*I  4  3

*3*I               *3*I  3  2  1      *3*I  1

----------------   ----------------   ----------------

*1*I  4            *1*I               *1*I

*2*I  2  1         *2*I  4  1         *2*I  4  3  2

*3*I  3            *3*I  3  2         *3*I  1

----------------   ----------------   ----------------

                                      *1*I

                                      *2*I  4  3  2  1

                                      *3*I

                                      ----------------
```

3.er cas, n = 5

```
*1*I  5  4  3  2  1      *1*I  5

*2*I                     *2*I  3  2  1

*3*I                     *3*I  4

----------------------   ----------------------

*1*I  5  4  3  2         *1*I  5

*2*I  1                  *2*I  3  2

*3*I                     *3*I  4  1

----------------------   ----------------------
```

```
*1*I  5  4  3
*2*I  1
*3*I  2
---------------------
*1*I  5  4  3
*2*I
*3*I  2  1
---------------------
*1*I  5  4
*2*I  3
*3*I  2  1
---------------------
*1*I  5  4  1
*2*I  3
*3*I  2
---------------------
*1*I  5  4  1
*2*I  3  2
*3*I
---------------------
*1*I  5  4
*2*I  3  2  1
*3*I
---------------------
```

```
*1*I  5  2
*2*I  3
*3*I  4  1
---------------------
*1*I  5  2  1
*2*I  3
*3*I  4
---------------------
*1*I  5  2  1
*2*I
*3*I  4  3
---------------------
*1*I  5  2
*2*I  1
*3*I  4  3
---------------------
*1*I  5
*2*I  1
*3*I  4  3  2
---------------------
*1*I  5
*2*I
*3*I  4  3  2  1
---------------------
```

```
*1*I  3
*2*I  5  2
*3*I  4  1
---------------------
*1*I  3  2
*2*I  5
*3*I  4  1
---------------------
*1*I  3  2  1
*2*I  5
*3*I  4
---------------------
```

```
*1*I  1
*2*I  5  4  3
*3*I  2
---------------------
*1*I  1
*2*I  5  4  3  2
*3*I
---------------------
*1*I
*2*I  5  4  3  2  1
*3*I
---------------------
```

```
*1*I
*2*I  5
*3*I  4  3  2  1
---------------------
*1*I  1
*2*I  5
*3*I  4  3  2
---------------------
*1*I  1
*2*I  5  2
*3*I  4  3
---------------------
*1*I
*2*I  5  2  1
*3*I  4  3
---------------------
*1*I  3
*2*I  5  2  1
*3*I  4
---------------------
```

```
*1*I  3  2  1
*2*I  5  4
*3*I
---------------------
*1*I  3  2
*2*I  5  4  1
*3*I
---------------------
*1*I  3
*2*I  5  4  1
*3*I  2
---------------------
*1*I  3
*2*I  5  4
*3*I  2  1
---------------------
*1*I
*2*I  5  4  3
*3*I  2  1
---------------------
```

LA TORRE DE NAHOI. APLICACION DE UN PROCEDIMIENTO RECURSIVO PARA LA SOLUCION DE UN PROBLEMA COMBINATORIO

JAUME RIBERA

Depto. de Estadística
Facultad de Informática de Barcelona.

En esta nota exponemos el concepto de recursividad para la búsqueda de soluciones de algunos problemas combinatorios, en contraposición a la búsqueda exhaustiva, estudiando a fondo el ejemplo de la Torre de Hanoi.

DESCRIPCION DEL PROBLEMA

El problema de la Torre de Hanoi se puede describir de la siguiente manera:

Tenemos tres clavijas, A, B, C, y n discos de n diferentes diámetros. Todos los discos tienen un agujero en el centro de manera que se pueden apilar en cualquiera de las clavijas. Inicialmente todos los discos están colocados en la clavija A por orden de tamaño, el más grande abajo y el más pequeño arriba.

El problema consiste en pasar todos los discos de la pila A a la B utilizando el clavo C como lugar de paso intermedio. Como restricciones de nuestro problema hay que tener en cuenta que en cada movimiento sólo se puede mover un disco de un clavo a otro y que no puede haber nunca un disco de diámetro superior encima de otro de diámetro inferior.

El nombre de este problema viene de la leyenda que explica que cerca de Hanoi hay unos monjes que trabajan para encontrar la solución con 64 discos y que el día que lo consigan se acabará el mundo. Si eso fuera verdad no tendríamos que preocuparnos en absoluto. Aún en el caso de que pudieran hacer movimientos perfectos, sin error, un movimiento por segundo, tardarían millones y millones de años para conseguirlo.

Nosotros simplificaremos el problema y le daremos solución cuando $n = 2, 3$, y un procedimiento general recursivo que nos permitirá encontrar la solución para cualquier n, demostrando, al mismo tiempo, inductivamente que la solución existe.

ARBOL DE ESTADOS

En nuestro caso cuando $n = 2$, el problema consiste en pasar de esta configuración

a esta otra:

Para representar los movimientos posibles que nos irán describiendo los estados, utilizaremos un árbol, donde los nudos representarán los estados y los arcos los movimientos.

Así, para $n = 2$, la fig. 3, nos muestra el árbol de exploración hasta un nivel de profundidad de 3 movimientos.

En este árbol podemos observar que a pesar de que figuran 25 nudos de estado, sólo hay 9 diferentes. Como el estado representa totalmente la configuración, y como los estados idénticos representan puntos de conocimiento idénticos, podremos doblar nuestro árbol en aquellos nudos donde el estado coincide. Esto nos lleva a representar la búsqueda mediante un gráfico.

Es el que aparece en la fig. 4.

El número de estados posibles crece desmesuradamente al aumentar n. Tendremos que buscar otro método para la solución de nuestro problema.

SOLUCION RECURSIVA

Si nuestro problema inicial con n discos lo subdividimos en tres pasos, tendremos una nueva visión del problema.

Si tenemos n discos en A y queremos pasarlos a B, eso equivale a:

i) Pasar n-1 discos de A a C
ii) Mover el disco n de A a B
iii) Pasar los n-1 discos de C a B

Con este nuevo punto de vista hemos reducido el problema de la Torre de Hanoi de n discos a dos veces el mismo problema con n-1 discos.

Si aplicamos recursivamente esta descomposición llegaremos a la simplificación final de resolver el problema de la Torre de Hanoi cuando $n = 1$, que es trivial, y sólo será necesario mover el disco de una clavija a la otra.

Veamos con detalle la aplicación de este algoritmo a la resolución del problema cuando $n = 3$.

La notación H(n, A, B) define el problema consistente en mover n discos de la pila A a la pila B con las restricciones de la Torre de Hanoi, y para $B \rightarrow A$ el movimiento del disco superior de la pila B a la pila A.

Así nuestro problema es H(3, A, B), que equivale a:

H(2, A, C)
$A \rightarrow B$
H(2, C, B)

Un paso posterior de descomposición nos da:

H(1, A, B)
$A \rightarrow C$
H(1, B, C)
$A \rightarrow B$
H(1, C, A)
$C \rightarrow B$
H(1, A, B)

Si tenemos en cuenta que H(1, X, Y) equivale a $X \rightarrow Y$, la solución al problema de la Torre de Hanoi H(3, A, B) es:

$A \rightarrow B$
$A \rightarrow C$
$B \rightarrow C$
$A \rightarrow B$
$C \rightarrow A$
$C \rightarrow B$
$A \rightarrow B$

Estos movimientos corresponden a la fig. 5.

El anexo 1 nos muestra un listado de los programas FORTRAN-V recursivo para la resolución del problema de la Torre de Hanoi para cualquier n y un ejemplo de ejecución cuando $n = 3, 4, 5$.

Fácilmente se demuestra que el número de movimientos para resolver el problema con n discos es $2^n - 1$, y tal como ya habíamos adelantado, para $n = 64$, el orden del número de movimientos es 10^{19}, y si lo consideramos en un intervalo de un segundo por movimiento, nos resulta un tiempo total del orden de $5.8494 * 10^{11}$ años.

Los tiempos de ejecución de los programas del anexo 1 son: