

TORRES DE HANOI

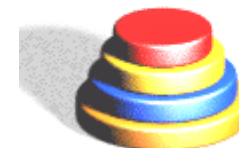


*Realizado por:
Jesús Ariza Pérez*

LEYENDA

Se cuenta que los monjes budistas mueven un disco cada segundo en una torre de sesenta y cuatro discos, cada uno de diferente tamaño. Por su fragilidad no puede colocarse un disco mas grande en uno mas pequeño. La leyenda dice que el mundo terminará antes que los sacerdotes terminen su trabajo.

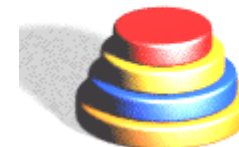
Ya que para resolver este problema en el numero mínimo de movimientos necesarios son $2^n - 1$, donde n es el número de discos. Los monjes tardarían algo menos de 585 mil millones de años en resolver el problema sin equivocarse ni una sola vez.



EXPLICACION

Este problema lógico consiste en cambiar una serie de discos, de una varilla inicial a una tercera varilla, pudiendo usar una segunda varilla intermedia para realizar los movimientos específicos. Eso si, debemos limitarnos a unos requisitos que dificultan el problema.

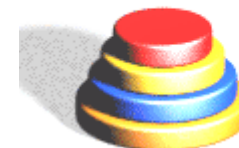
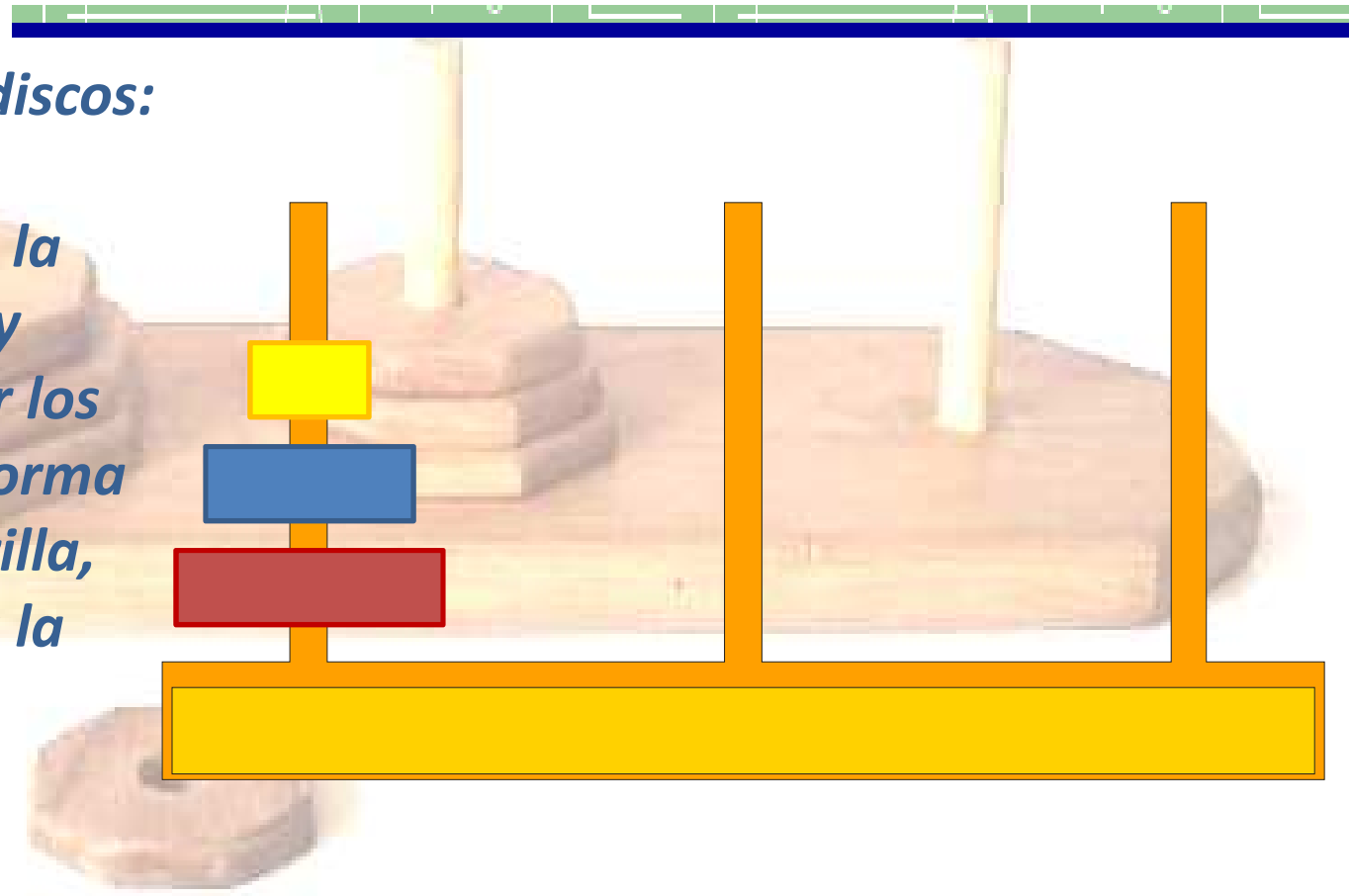
- Un disco de mayor tamaño no puede ir encima de uno mas pequeño.*
- Solo se pueden mover los discos de uno en uno.*
- Se saca el disco que esta colocado mas arriba de la varilla.*



EXPLICACION

Ejemplo para 3 discos:

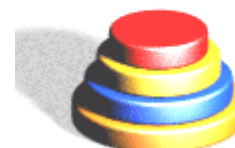
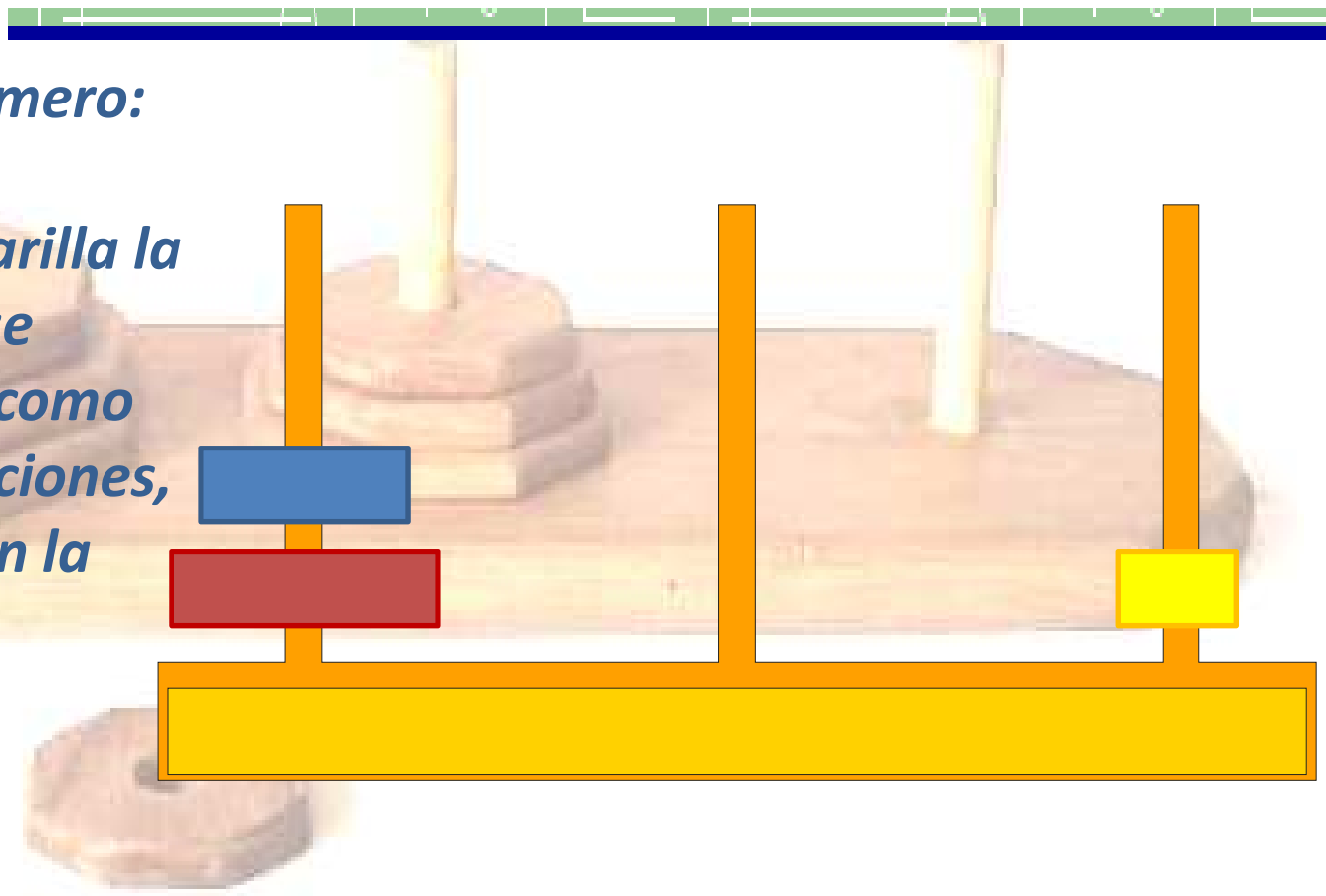
Comenzamos en la primera varilla, y debemos colocar los discos de igual forma en la tercera varilla, la mas situada a la derecha.



EXPLICACION

-Movimiento primero:

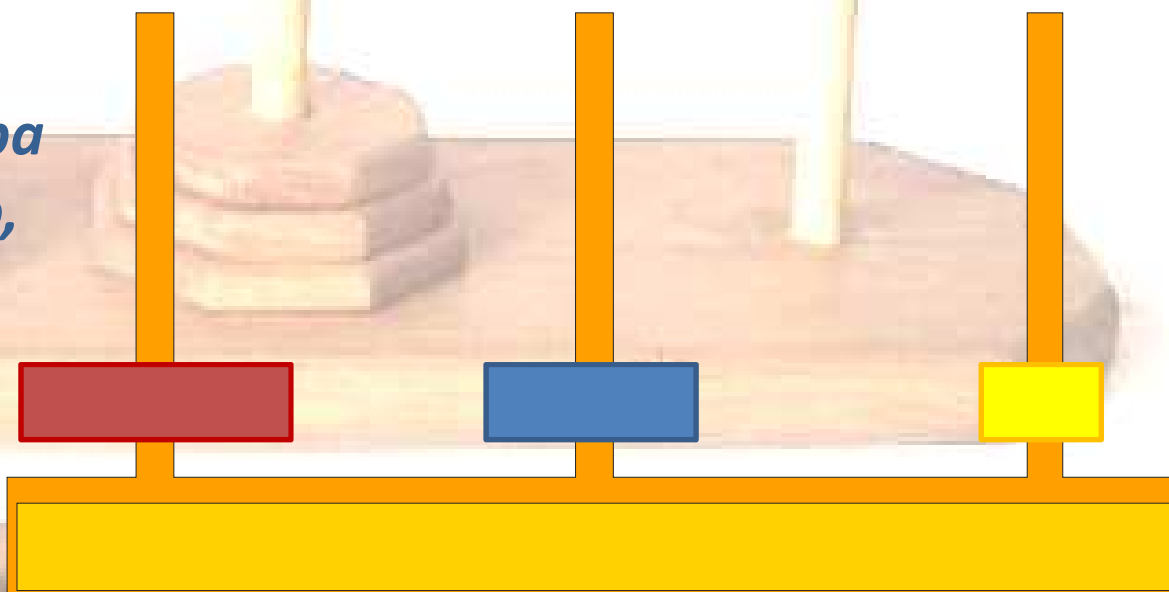
Sacamos de la varilla la que mas arriba se encuentra, tal y como explica las condiciones, y la colocamos en la tercera varilla.



EXPLICACION

-Movimiento segundo:

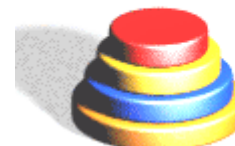
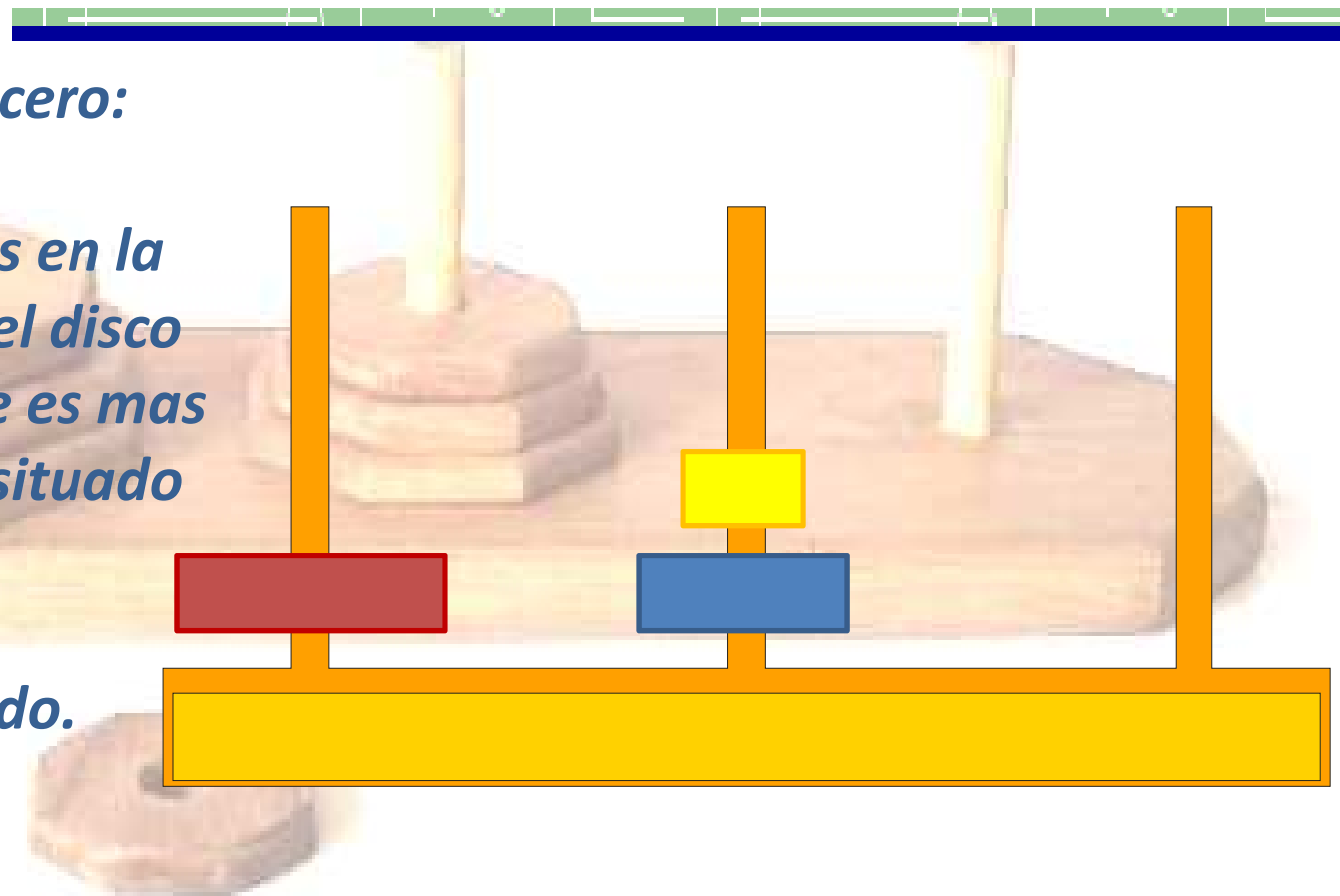
Sacamos de la varilla inicial la que mas arriba se encuentra de nuevo, y la colocamos en la segunda varilla, esta vez no podemos colocarla en la tercera porque es mayor que la que ya se encuentra en ella.



EXPLICACION

-Movimiento tercero:

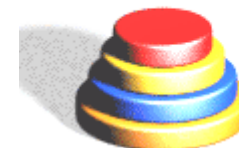
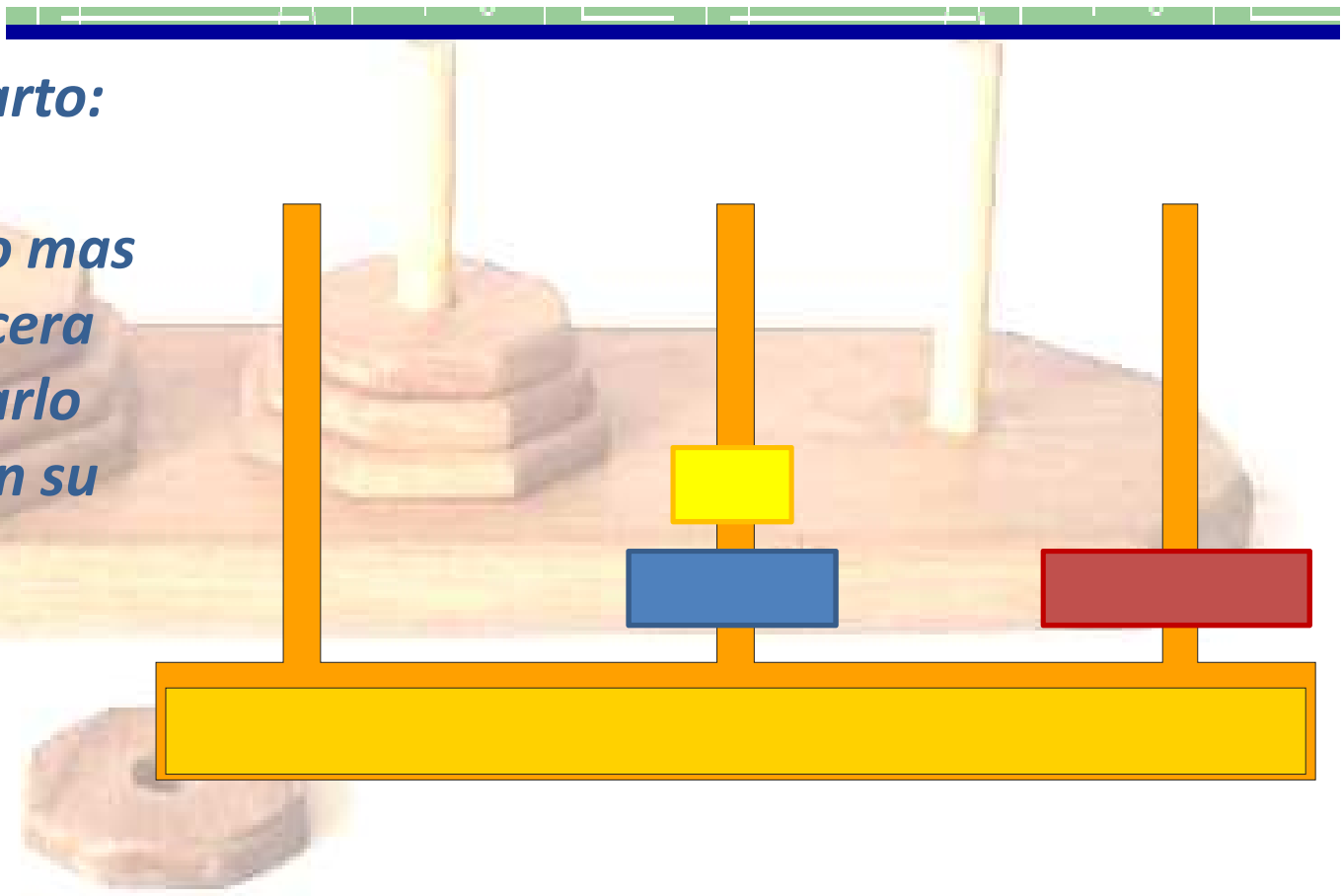
Ahora colocamos en la segunda varilla el disco de la tercera que es mas pequeño que el situado en la segunda, por tanto es un movimiento válido.



EXPLICACION

-Movimiento cuarto:

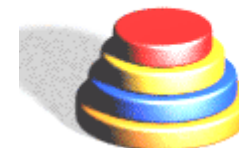
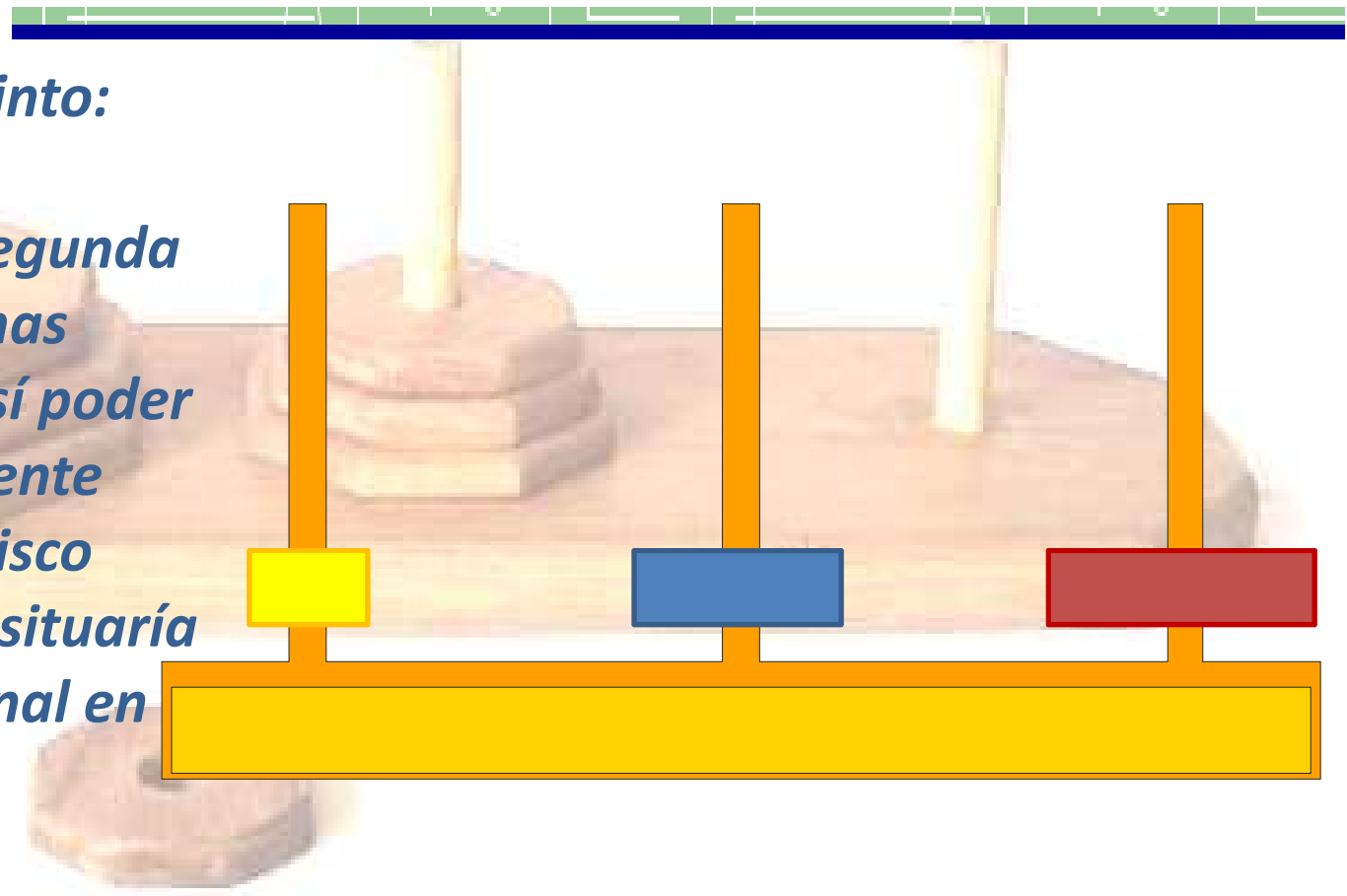
Ponemos el disco mas grande en la tercera varilla para situarlo correctamente en su posición final.



EXPLICACION

-Movimiento quinto:

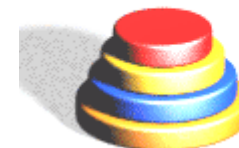
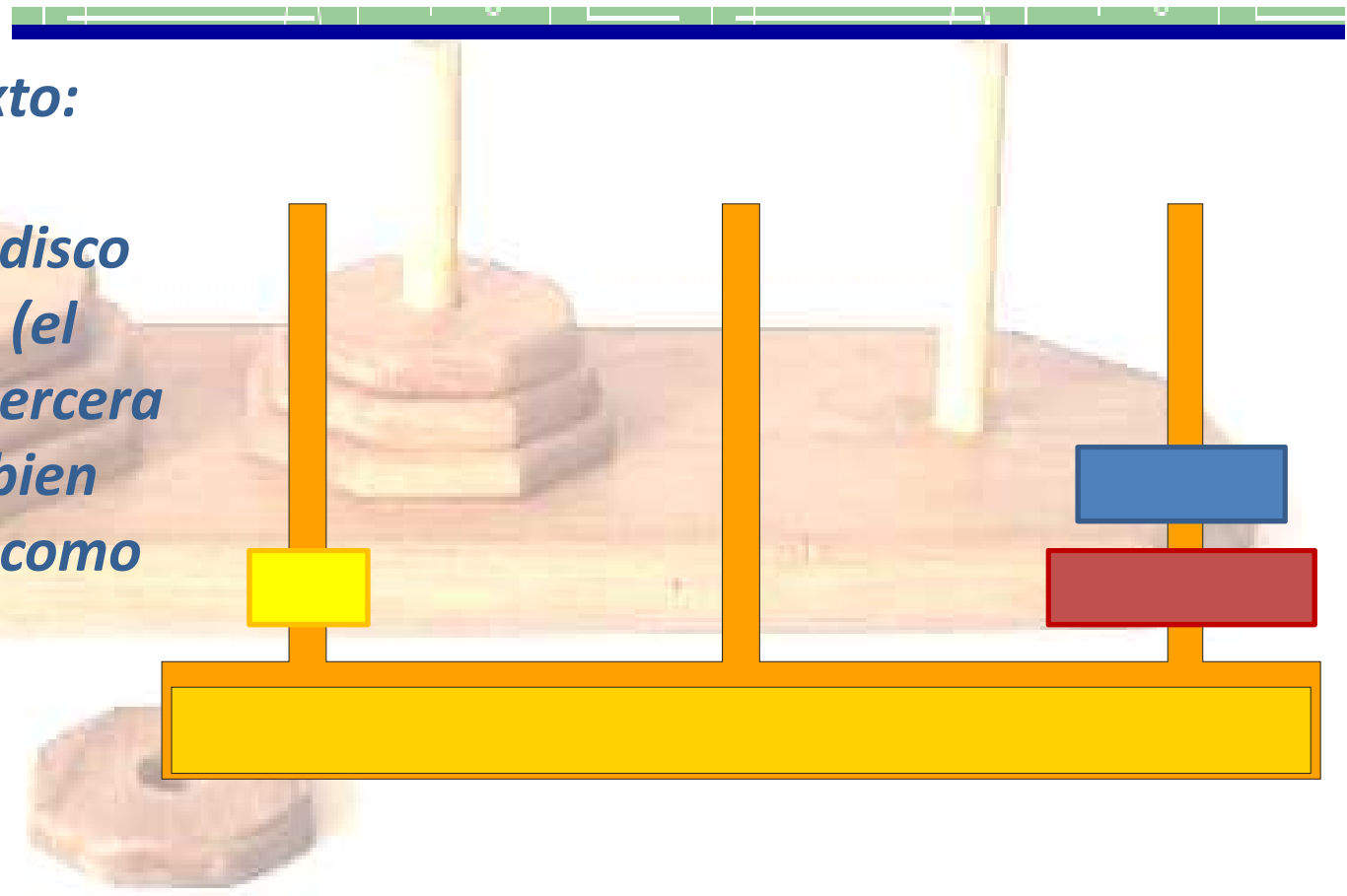
Sacamos de la segunda varilla el disco mas pequeño para así poder sacar en el siguiente movimiento el disco mediano que se situaría en su posición final en tercera varilla.



EXPLICACION

-Movimiento sexto:

Ya colocamos el disco correspondiente (el mediado) en la tercera varilla, que también será su posición como solución final.

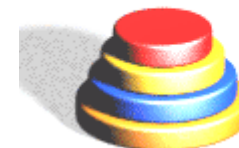
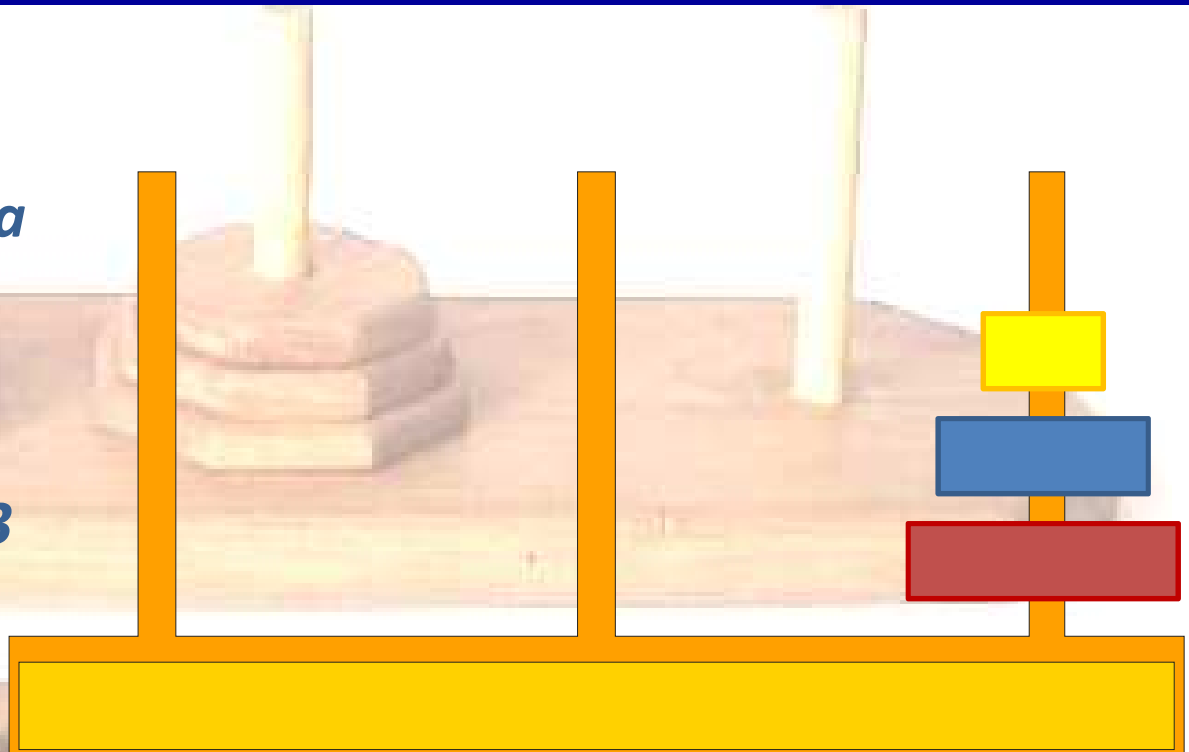


EXPLICACION

-Movimiento séptimo:

Movemos el disco de la primera varilla a la última. Y con esto tenemos una solución para mi problema de 3 discos, que además es solución mínima.

Hemos necesitado 7 movimientos para sólo 3 discos.



MODELO

Minimizar:

$$\sum X_i;$$

Sujeto a:

$$X_i + X(i+1) = X?; //X? \text{ condición entrada}$$

$$X(i+1) + X(i+2) \leq 2;$$

$$X_i + X(i+2) \leq 2;$$

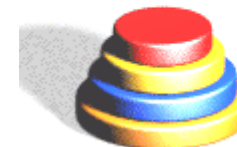
//Si $n > 1$ para cada paquete de puentes.

$$X(i+9) = X_i + X(i+2)$$

$$X(i+10) = X? - X(i+9)$$

$$X(i+11) = X(i+9)$$

Donde $X_i \leq 1$;

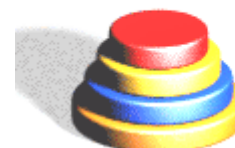
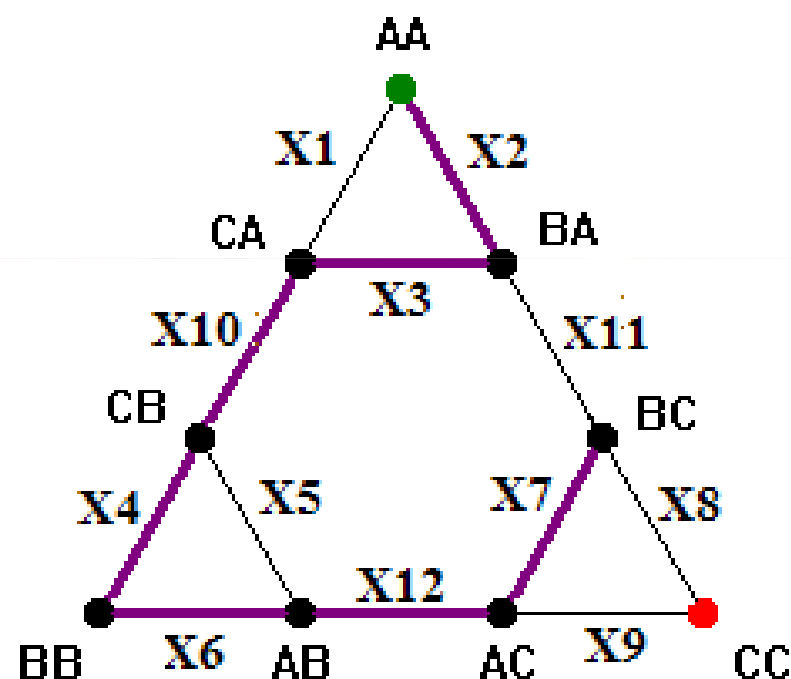


MODELO

*Diagrama de estados
(ejemplo para 2 discos):*

*AA punto inicial.
CC punto final.*

*Cada arco se enumera,
vale 1 si accedemos por
él y 0 sino lo
recorremos.*

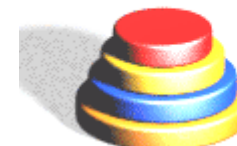


MODELO LP PARA 2 DISCOS

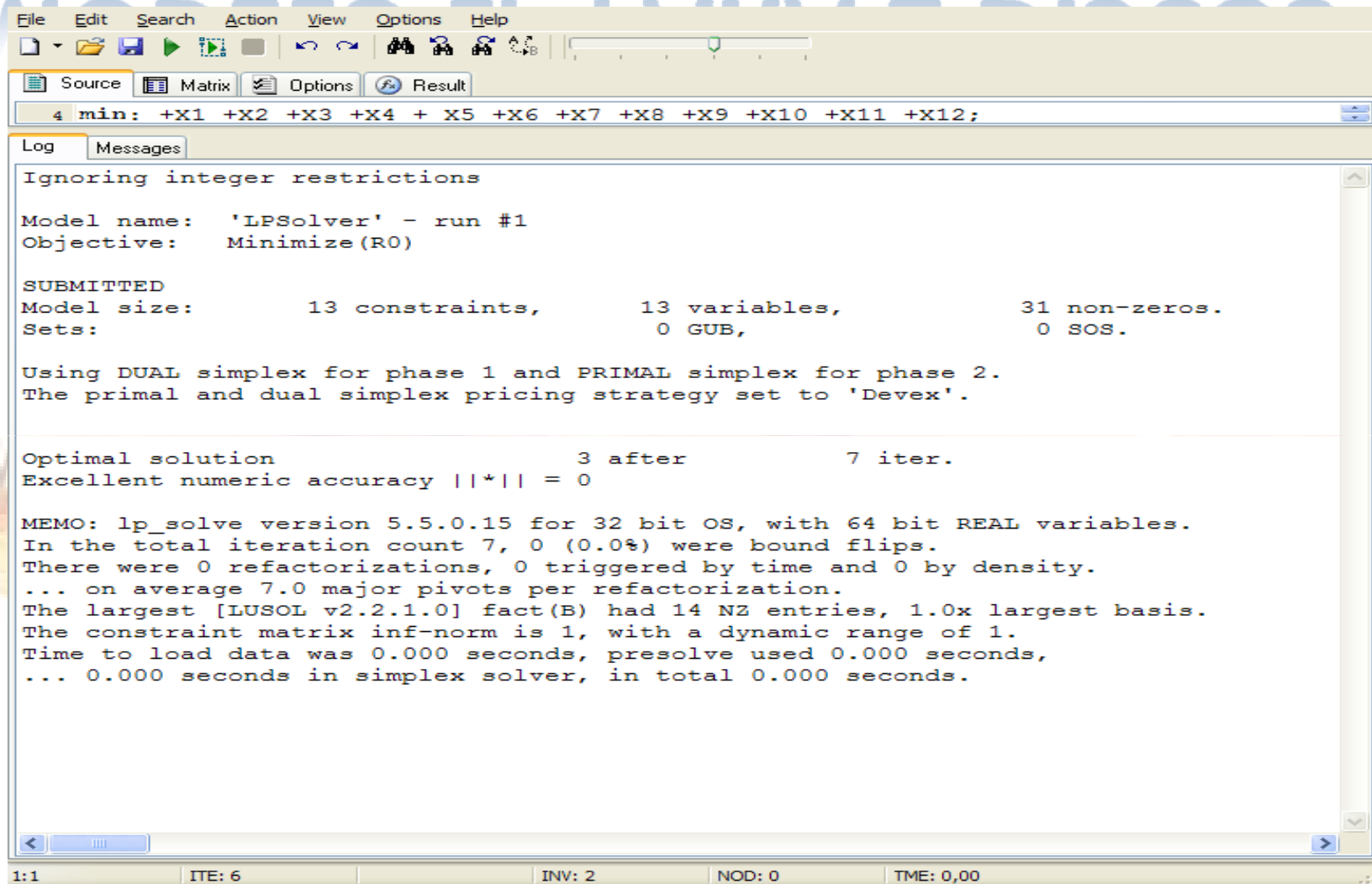
```

File Edit Search Action View Options Help
Source Matrix Options Result
1 /* Hanoi para n=2*/
2
3 /* Objective function */
4 min: +X1 +X2 +X3 +X4 + X5 +X6 +X7 +X8 +X9 +X10 +X11 +X12;
5
6 /* Constraints */
7
8 COND1: X0 =1; /*condicion de entrada*/
9
10 COND2: +X1 + X2 = +X0;
11 COND3: +X2 + X3 <=2;
12 COND4: +X1 +X3 <=2;
13
14
15 COND5: +X4+X5=+X10;
16 COND6: +X5+X6 <=2;
17 COND7: +X4+X6 <=2;
18
19 COND8: +X7+X8 =+X11+X12;
20 COND9: +X8+X9 <=2;
21 COND10: +X7+X9 <=2;
22
23
24 paso1: +X10=+X1+X3;
25 paso2: +X11=X0-X10;
26 paso3: +X12=+X10;
27
28
29
30 /* Variable bounds */
31 X1 <= 1;
32 X2 <= 1;
33 X3 <= 1.
1:1

```



MODELO LP PARA 2 DISCOS



The screenshot shows the LPSolver software interface. The 'Source' tab is active, displaying the following code:

```
4 min: +X1 +X2 +X3 +X4 + X5 +X6 +X7 +X8 +X9 +X10 +X11 +X12;
```

The 'Log' tab is also visible, showing the following output:

```
Ignoring integer restrictions

Model name: 'LPSolver' - run #1
Objective: Minimize(R0)

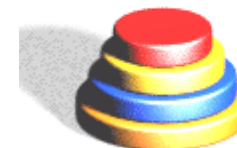
SUBMITTED
Model size:      13 constraints,      13 variables,      31 non-zeros.
Sets:           0 GUB,                0 SOS.

Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.

Optimal solution          3 after          7 iter.
Excellent numeric accuracy ||*|| = 0

MEMO: lp_solve version 5.5.0.15 for 32 bit OS, with 64 bit REAL variables.
In the total iteration count 7, 0 (0.0%) were bound flips.
There were 0 refactorizations, 0 triggered by time and 0 by density.
... on average 7.0 major pivots per refactorization.
The largest [LUSOL v2.2.1.0] fact(B) had 14 NZ entries, 1.0x largest basis.
The constraint matrix inf-norm is 1, with a dynamic range of 1.
Time to load data was 0.000 seconds, presolve used 0.000 seconds,
... 0.000 seconds in simplex solver, in total 0.000 seconds.
```

At the bottom of the window, the status bar shows: 1:1 | ITE: 6 | INV: 2 | NOD: 0 | TME: 0,00



BIBLIOGRAFIA

-<http://www.psicoactiva.com/>, 1998

-(Wikipedia 2009)

http://es.wikipedia.org/wiki/Torres_de_Han%C3%B3i

-<http://thales.cica.es/rd/Recursos/rd99/ed99-0222-02/matematicas1.htm>, 2006

-Juego interactivo:

<http://www.mazeworks.com/hanoi/index.htm> , 2002

