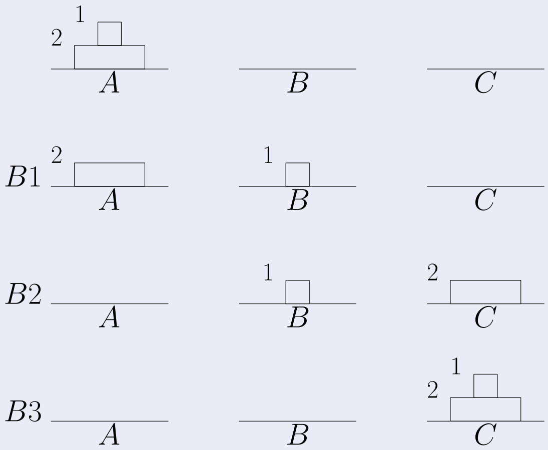
Αλγόριθμοι και Πολυπλοκότητα

Καθηγητής Ν.Μισυρλής

23 Φεβρουαρίου 2018

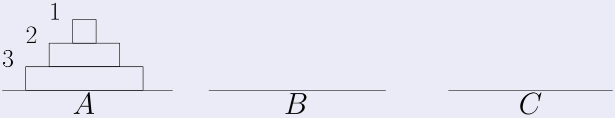
Hanoi

2 Δίσκοι - Μεταφορά $A \rightarrow C$

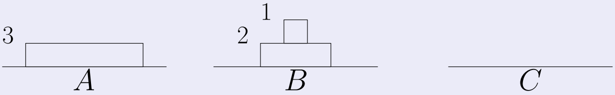


Hanoi

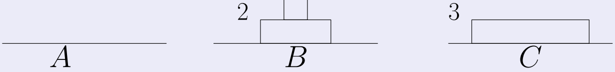
3 Δίσκοι - Μεταφορά $A \to C$



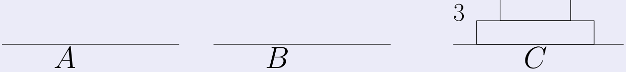
$B1 : A \xrightarrow{C} B$



$B2 : A \to C$

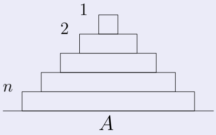


$B2 : B \xrightarrow{A} C$

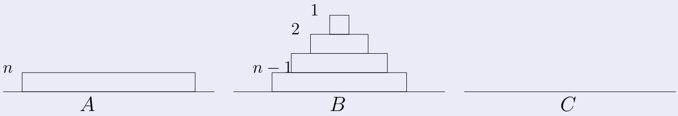


Hanoi

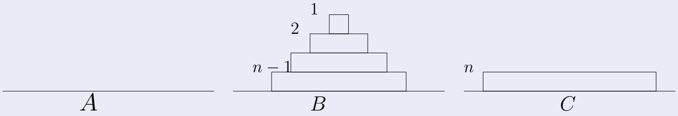
Γενικά n Δίσκοι. Μεταφορά $A \longrightarrow C$



B1: Μεταφορά $n - 1$ Δίσκων $A \xrightarrow{C} B$

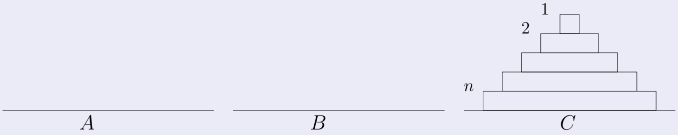


B2: Μεταφορά του n Δίσκου $A \longrightarrow C$



Hanoi

B3: Μεταφορά των $n-1$ Δίσκων $B \rightarrow C$ με τη χρήση του A



Hanoi

Αναδρομικός Αλγόριθμος

Hanoi(n, A, B, C) /* Μεταφορά $A \rightarrow C$ με τη χρήση του B */

Αν $n \geq 1$

  Hanoi(n-1, A, C, B) /* Μεταφορά $A \rightarrow B$ με τη χρήση του C */

  Μεταφορά του n δίσκου από τον A στύλο στον C

  Hanoi(n-1, B, A, C) /* Μεταφορά $B \rightarrow C$ με τη χρήση του A */

Hanoi

Πολυπλοκότητα

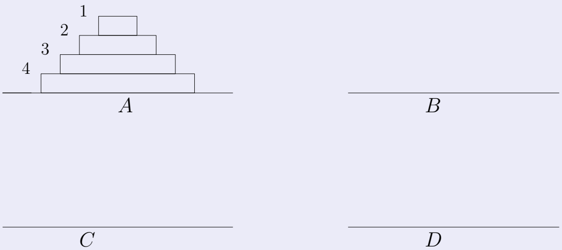
$$
\begin{aligned}
T(n) &= 2T(n-1)+1 \\
&= 2[2T(n-2)+1]+1 \\
&= 2^2T(n-2)+2+1 \\
&= 2[2^2T(n-3)+1]+2+1 \\
&= 2^3T(n-3)+2^2+2^1+2^0 \\
&\;\;\vdots \\
&= 2^kT(n-k)+2^{k-1}+2^{k-2}+\cdots+2^1+2^0 \\
&\;\;\vdots \\
&= 2^{n-1}T(1)+2^{n-2}+\cdots+2^1+2^0 \\
T(n) &= 2^{n-1}+\cdots+2^1+2^0 = \frac{2^{n-1}\cdot 2-2^0}{2-1} = 2^n-1
\end{aligned}
$$

Άρα

$$T(n) = \Theta(2^n)$$

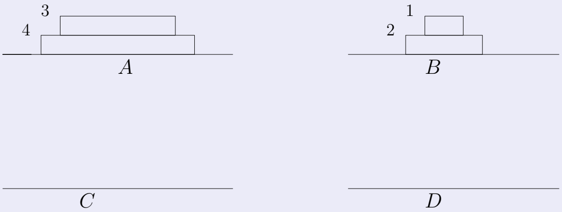
Hanoi-4 Στύλοι

Μεταφορά $A \rightarrow D$



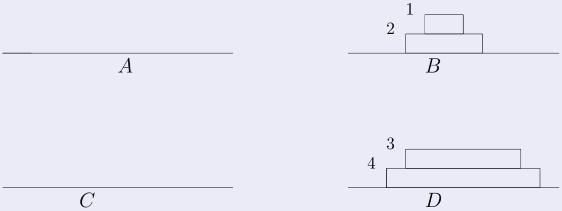
Hanoi-4 Στύλοι

B1: Μεταφορά των δύο πάνω δίσκων $A \rightarrow B$ με τη χρήση του C ή D



Hanoi-4 Στύλοι

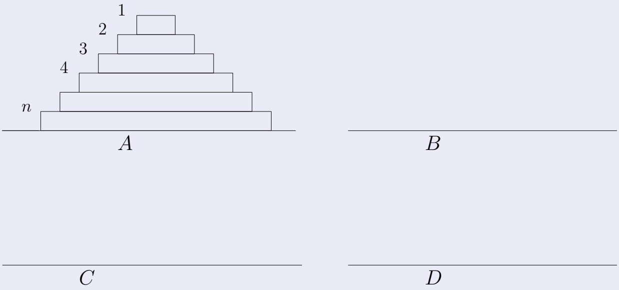
B2: Μεταφορά των δύο δίσκων $A \rightarrow D$ με τη χρήση του C



B3: Μεταφορά των δύο δίσκων $B \rightarrow D$ με τη χρήση του A ή C

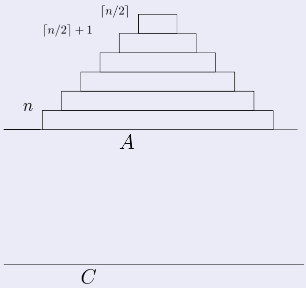
Hanoi-4 Στύλοι

Γενικά: Μεταφορά n δίσκων $A \rightarrow D$ με τη χρήση των B και C



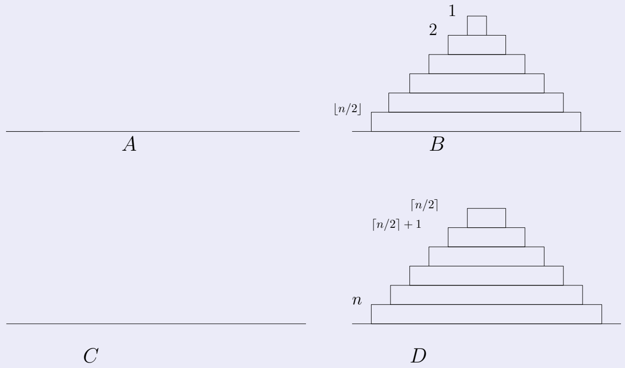
Hanoi-4 Στύλοι

B1: Μεταφορά των πρώτων $\lfloor n/2 \rfloor$ δίσκων $A \rightarrow B$ με τη χρήση του C ή D



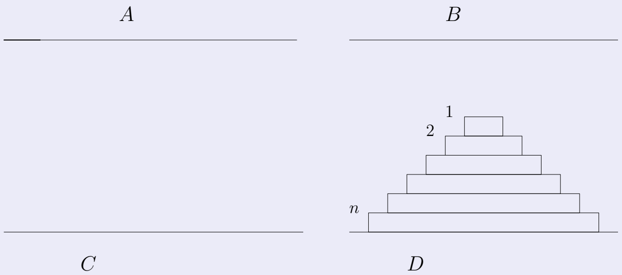
Hanoi-4 Στύλοι

B2: Μεταφορά $\lceil n/2 \rceil$ δίσκων $A \rightarrow D$ με τη χρήση του C



Hanoi-4 Στύλοι

B3: Μεταφορά $\lfloor n/2 \rfloor$ δίσκων $B \rightarrow D$ με τη χρήση του C ή A



Hanoi-4 Στύλοι

Αναδρομικός Αλγόριθμος

Hanoi(n, A, B, C) /* $A \rightarrow D$ */

Αν $n \geq 1$

Hanoi($\lfloor n/2 \rfloor$, A, C, D, B) /* $A \rightarrow B$ με τη χρήση του C ή D */

Hanoi($\lceil n/2 \rceil$, A, B, C, D) /* $A \rightarrow D$ με τη χρήση του C γιατί ο B είναι κατειλημμένος*/

Hanoi($\lfloor n/2 \rfloor$, B, A, C, D) /* $B \rightarrow D$ με τη χρήση του A ή C */

Hanoi-4 Στύλοι

Πολυπλοκότητα

$$
\begin{aligned}
T(n) &= 3T(n/2) \\
&= 3[3T(n/2^2)] \\
&= 3^2T(n/2^2) \\
&\vdots \\
&= 3^kT(n/2^k)
\end{aligned}
$$

Άρα

$$\frac{n}{2^k} = 1 \Rightarrow k = logn$$

$$3^k = 3^{log_2 n} = n^{log_2 3} = n^{1.59}$$

$$T(n) = 3^kT(1) = 3^k \cdot 1 = n^{log_2 3} = n^{1.59}$$

$$T(n) = \Theta(n^{1.59})!$$

Hanoi

5 Στύλοι- Πολυπλοκότητα

$$\begin{aligned} T(n) &= 4T(n/3) \\ &\vdots \\ &= 4^k T(n/3^k) \\ \frac{n}{3^k} = 1 \text{ ή } k = log_3 n &= 4^k = 4^{log_3 n} = n^{log_3 4} = n^{1.26} \end{aligned}$$

6 Στύλοι- Πολυπλοκότητα

$$\begin{aligned} T(n) &= 5T(n/4) = \cdots = 5^k T(n/4^k) \\ (k = log_4 n) &= 5^k = 5^{log_4 n} = n^{log_4 5} = n^{1.16} \end{aligned}$$

Hanoi

7 Στύλοι- Πολυπλοκότητα

$$T(n) = \quad 6T(n/5) = n^{log_5 6} = n^{1.11}$$

8 Στύλοι- Πολυπλοκότητα

$$T(n) = \quad 7T(n/6) = n^{log_6 7} = n^{1.08}$$

$$T(n) = O(n^{1.08})$$

Απαιτούνται 8 στύλοι προκειμένου να έχουμε γραμμική πολυπλοκότητα.