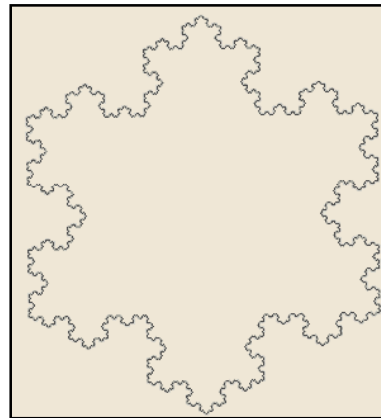
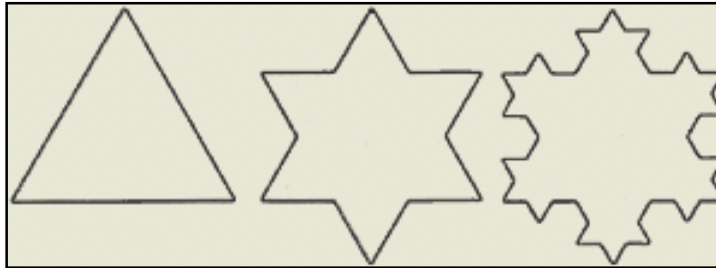


ΑΝΑΔΡΟΜΗ

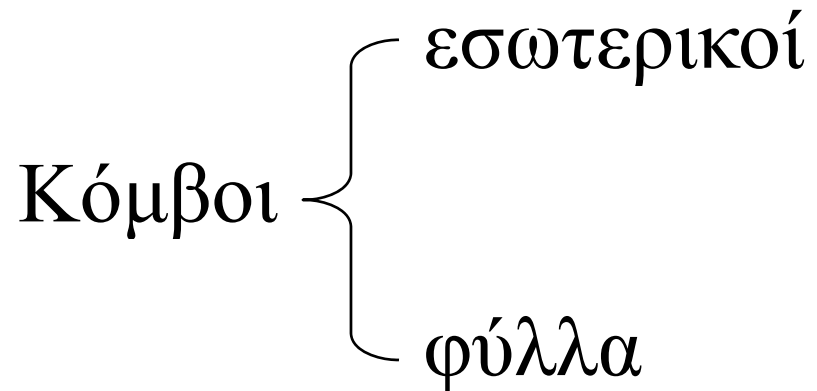


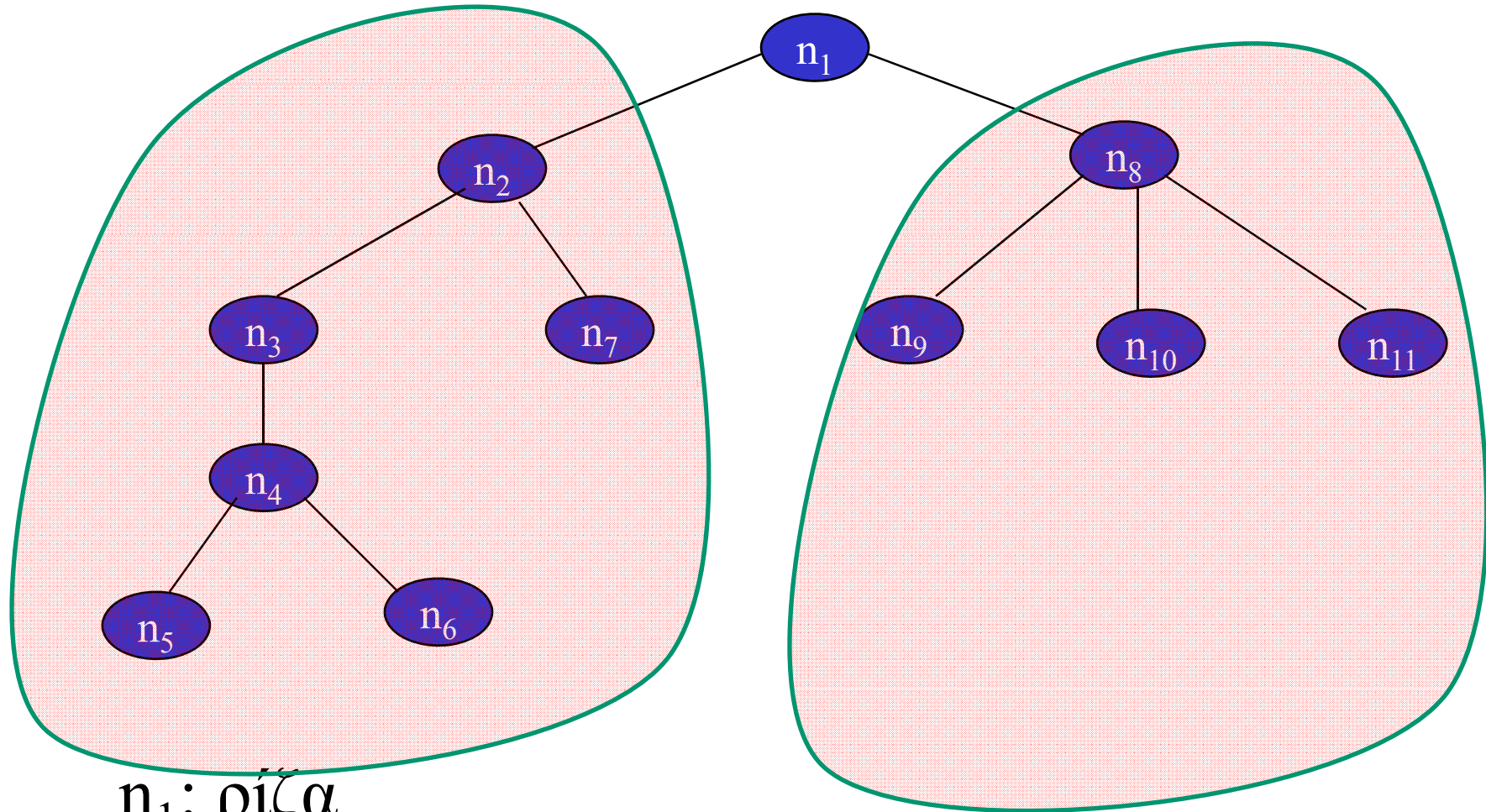
Νιφάδες του Von Koch

ΑΝΑΔΡΟΜΗ

- Εργαλείο ορισμού ενός τύπου
- Αλγοριθμική μέθοδος
- Τεχνική προγραμματισμού
 - πιο αισθητική
 - πιο φυσική
 - πιο ισχυρή

Ένα δένδρο είναι είτε ένα ατομικό δένδρο (ένα φύλλο), είτε ένας κόμβος και μια ακολουθία από δένδρα.





n_1 : ρίζα

$n_5, n_6, n_7, n_9, n_{10}, n_{11}$: φύλλα

n_1, n_2, n_3, n_4, n_5 : εσωτερικοί κόμβοι

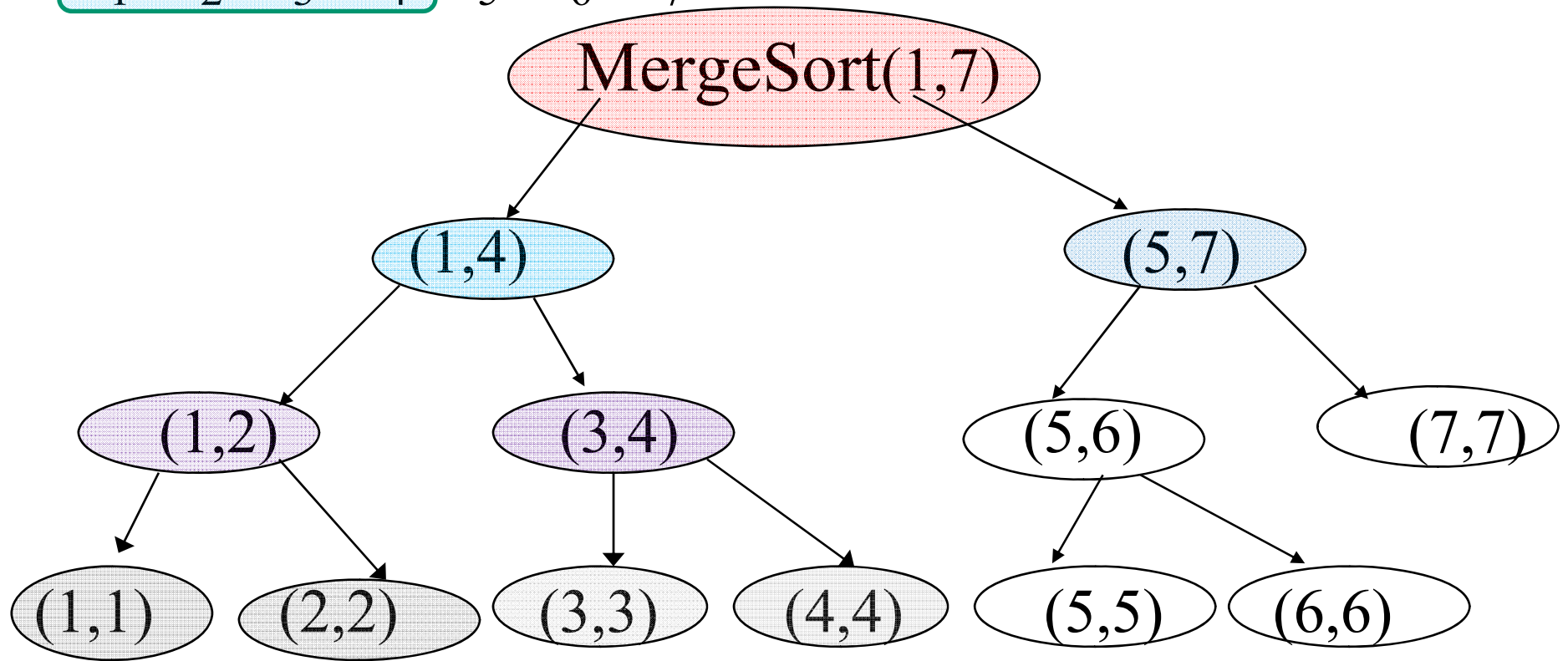
Divide And Conquer

- **Διαίρεση** του προβλήματος σε μικρότερα προβλήματα
- **Επίλυση** των μικροτέρων προβλημάτων
- **Συνδυασμός** λύσεων για την εύρεση της αρχικής λύσης

Merge Sort

- **Divide:** πίνακας με n στοιχεία $\rightarrow 2$ υποπίνακες με $\lfloor n/2 \rfloor$ και $\lceil n/2 \rceil$ στοιχεία
- **Conquer:** ταξινόμησε τους δυο υποπίνακες αναδρομικά με Merge Sort
- **Combine:** Merge τους δυο ταξινομημένους πίνακες

A: $6_1 \ 9_2 \ 2_3 \ 7_4 \ 4_5 \ 5_6 \ 8_7$



➤ 2 6 7 9 || 4 5 8

➤ 2 6 7 9 || 8 5 4 → 2 4 5 6 7 8 9

Merge 2 ταξ/νες ακολουθίες

Merge (A,f,q,r)

```
όρισε πίνακες a[1..q-f+2] και b[1..r-q+1]  
for i=1 to q-f+1 do a[i]=A[f+i-1]  
for j=1 to r-q do b[j]=A[q+j]  
a[q-f+2]=∞; b[r-q+1]=∞
```

```
i=1; j=1;  
for k=f to r do  
  if a[i] <= b[j] then A[k]=a[i]; i=i+1  
  else A[k]=b[j]; j=j+1
```

Παράδειγμα: A: 2 6 7 9 4 5 8 (είσοδος), a: 2 6 7 9 ∞
b: 4 5 8 ∞

A: 2 4 5 6 7 8 9 (έξοδος)

Πολυπλοκότητα: O(n)

Merge Sort (*****)

- **MergeSort** (A,l,r)

If $l < r$ then $q = \lfloor (l+r)/2 \rfloor$

MergeSort (A,l,q)

MergeSort (A,q+1,r)

Merge (A,l,q,r)

end if

Πολυπλοκότητα MergeSort

- $T(n) = \Theta(1)$ if $n=1$
- $T(n) = 2T(n/2) + \Theta(n)$ if $n > 1$
- $\Theta(n \log n)$

$$T(n) = \begin{cases} 1 & \text{αν } n=1 \\ aT(n/b) + d(n) & \end{cases}$$

$(n = b^k$, αν όχι τότε upper bound της $T(n)$)

MergeSort: $a = b = 2$, $d(n) = \Theta(n)$

Επίλυση της Γενικής περίπτωσης (Substitution method)

Γενική μορφή στο i βήμα: $T(n/b^i) = a T(n/b^{i+1}) + d(n/b^i)$

$$T(n) = aT(n/b) + d(n)$$

$$= a[aT(n/b^2) + d(n/b)] + d(n)$$

$$= a^2T(n/b^2) + ad(n/b) + d(n)$$

$$= a^2[aT(n/b^3) + d(n/b^2)] + ad(n/b) + d(n)$$

$$= a^3T(n/b^3) + a^2d(n/b^2) + ad(n/b) + d(n)$$

=

$$= a^i T(n/b^i) + \sum_{j=0}^{i-1} a^j d(n/b^j)$$

Επίλυση της Γενικής περίπτωσης (Substitution method)

$$T(n) = a^i T(n/b^i) + \sum_{j=0}^{i-1} a^j d(n/b^j)$$

Αν υποθέσουμε $n = b^k \Rightarrow T(n/b^k) = T(1) = 1$. Για $i = k$ έχουμε:

$$T(n) = a^k + \sum_{j=0}^{k-1} a^j d(b^{k-j})$$

Επίσης $k = \log_b n$ και επομένως $a^k = a^{\log_b n} = n^{\log_b a}$
(constant power)

Ειδική περίπτωση: Mergesort

$$T(n) = a^k + \sum_{j=0}^{k-1} a^j d(b^{k-j}) = n^{\log_b a} + \sum_{j=0}^{k-1} a^j d(b^{k-j})$$

$$d(n) = (n-1), a=b=2 :$$

$$\begin{aligned} T(n) &= n + \sum_{j=0}^{k-1} 2^j (2^{k-j} - 1) = n + \sum_{j=0}^{k-1} 2^k - 2^j = n + 2^k k - (2^k - 1) / (2 - 1) \\ &= \cancel{n} + n \log n - \cancel{n} + 1 \\ &= n \log n + 1 \end{aligned}$$

Οι πύργοι του Hanoi

- 3 στύλοι: 1, 2, 3
 - n δίσκοι στο στύλο 1, σχηματίζοντας κώνο
-
- Να μεταφερθούν όλοι από τον 1 στον 3
 - μόνο ένα δίσκος κάθε φορά
 - να μην είναι ποτέ ένας μικρός κάτω από ένα μεγαλύτερο

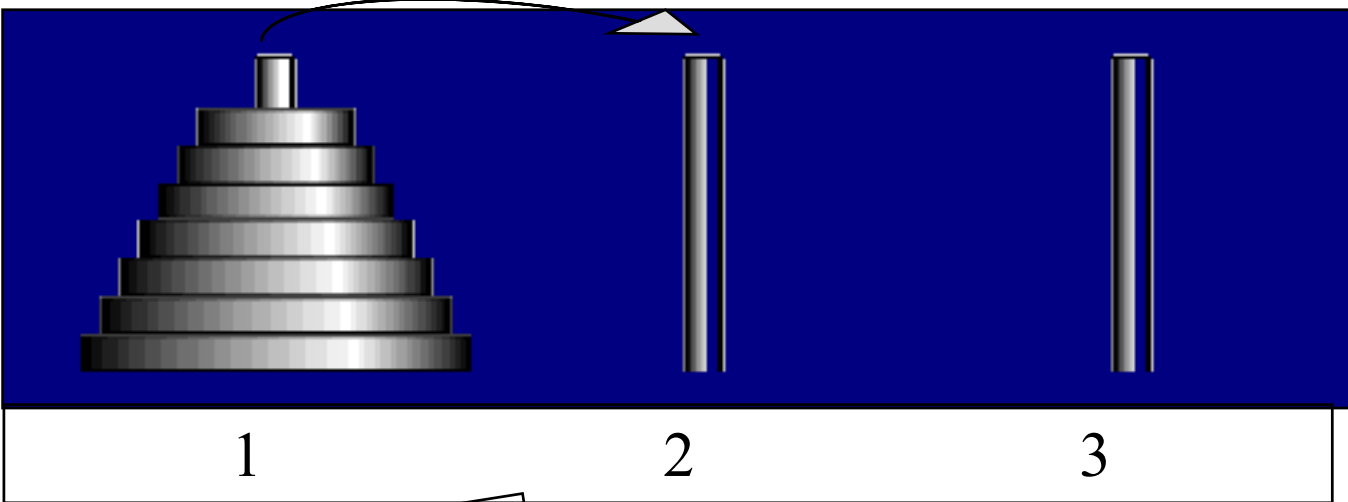
Οι πύργοι του Hanoi: Αλγόριθμος

- Αν $n < 1 \rightarrow \text{ok}$
- Εστω το πρόβλημα έχει λυθεί για $n-1$ δίσκους

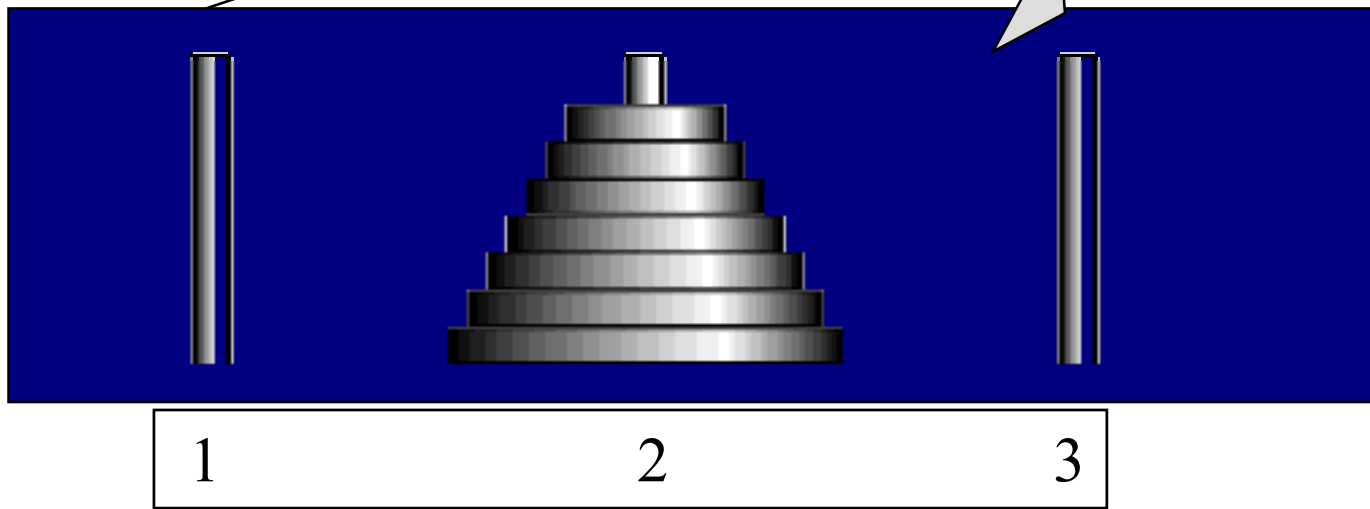
Λύση για n δίσκους (από $i \rightarrow j$)

1. Μεταφέρουμε τους $n-1$ δίσκους από τον i στον $k = 6-(i+j)$
2. Παίρνουμε το μεγάλο δίσκο από τον i και τον τοποθετούμε στον j
3. Μεταφέρουμε τους $n-1$ δίσκους από τον k στον j

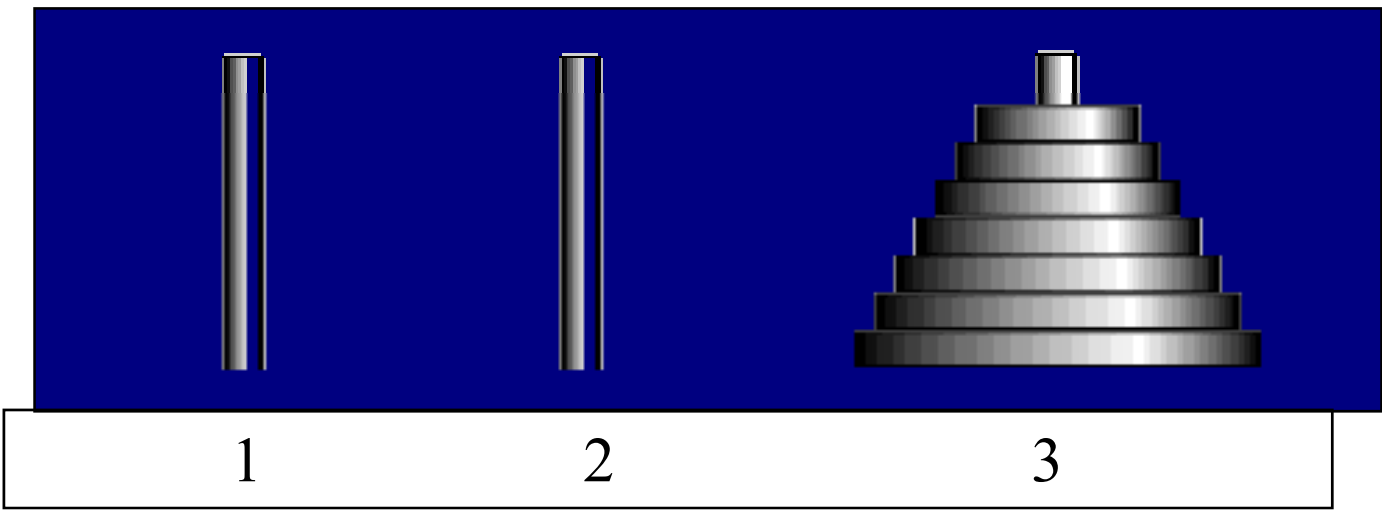
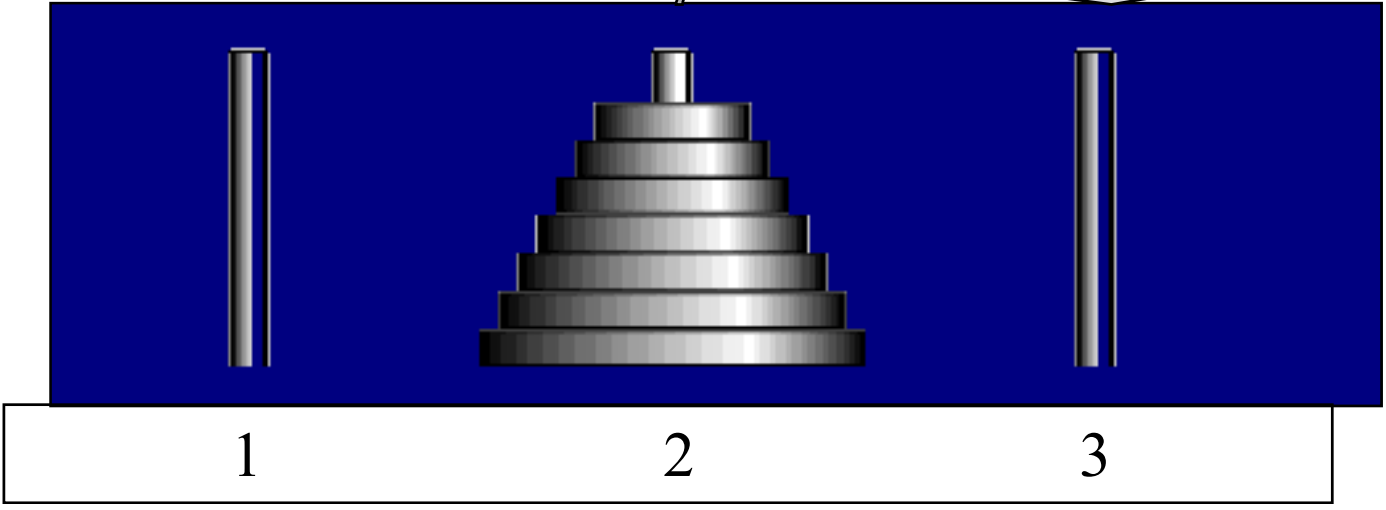
Hanoi(n-1,i,6-(i+j))



Από i → j



Hanoi(n-1,6-(i+j),j)



Οι πύργοι του Hanoi: Αλγόριθμος

```
Hanoi(n:integer; i, j:integer);
```

```
if n > 0 then
```

```
    Hanoi(n-1, i, 6-(i+j))
```

Μετέφερε 1 δίσκο απο τον i στον j

```
    Hanoi(n-1, 6-(i+j), j)
```

```
endif
```

Πύργοι του Hanoi

Πολυπλοκότητα:

$$T_n = 2T_{n-1} + 1 \quad (n \geq 1)$$

$$T_0 = 1$$

$$T_n = 2T_{n-1} + 1 \quad (\times 2^0)$$

$$T_{n-1} = 2T_{n-2} + 1 \quad (\times 2^1)$$

$$T_{n-2} = 2T_{n-3} + 1 \quad (\times 2^2)$$

....

$$T_2 = 2T_1 + 1 \quad (\times 2^{n-2})$$

$$T_1 = 2T_0 + 1 \quad (\times 2^{n-1})$$

$$T_n = 2^n T_0 + (1 + \dots + 2^{n-1})$$

μέθοδος των αθροισμένων

$$T_n = 2^n \times 1 + (2^n - 1) / (2 - 1) \Rightarrow T_n = 2^{n+1} - 1, \quad n \geq 0$$

παραγόντων

Άσκηση:

$$T(n) = T(n-1) + 2, \quad T(1) = 1$$

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

.....

.....

$$= 2 + \dots + 2 + T(1)$$

$\underbrace{\hspace{1.5cm}}_{n-1 \text{ φορές}}$

$$T(n) = (n-1)2 + 1 \leq 2n$$

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