Παράδειγμα βέλτιστου αλγορίθμου. Δομή "σωρού" και αλγόριθμος Heapsort για διάταξη λίστας

Δομή σωρού είναι ένα δέντρο που προκύπτει από ένα πλήρες δυαδικό δέντρο με ρίζα, όταν του αφαιρέσουμε μερικούς κόμβους (ή κανένα) από τα δεξιά του τελευταίου επιπέδου. Πιο συγκεκριμένα δομή σωρού έχουμε όταν στο δέντρο

(α) όλοι οι εσωτερικοί κόμβοι έχουν δύο παιδιά, με μόνη πιθανή εξαίρεση αυτή που αναφέρεται παρακάτω στο (γ),

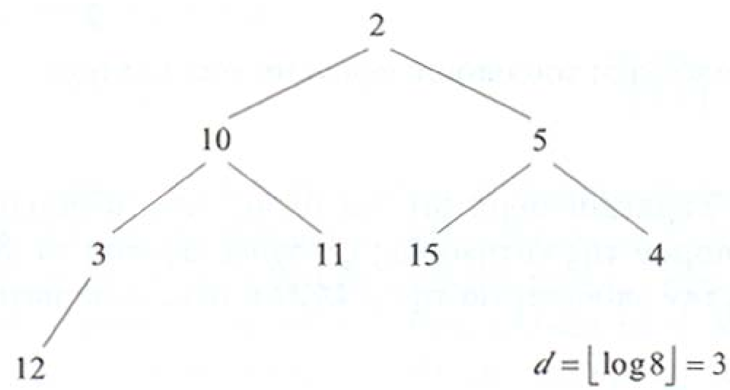
(β) στο επίπεδο *d-1* τα φύλλα είναι όλα στα δεξιά των εσωτερικών κόμβων, και

(γ) ο πιο δεξιά εσωτερικός κόμβος του επιπέδου *d-1* επιτρέπεται να έχει βαθμό 1, οπότε θα έχει μόνο αριστερό παιδί.

Σωρός (Heap) είναι ένα δέντρο, στους κόμβους του οποίου αντιστοιχούν αντιπρόσωποι από μια λίστα *L* και το οποίο είναι δομή σωρού με την εξής επιπλέον ιδιότητα: ο αντιπρόσωπος κάθε κόμβου δεν είναι μικρότερος των αντιπροσώπων των παιδιών του, αν ο κόμβος έχει παιδιά.

L $\Leftrightarrow$ ΔΟΜΗ HEAP

	L
$2^0 = 1$	2
$2^1 = 2$	10
3	5
$2^2 = 4$	3
5	11
6	15
7	4
$2^3 = n = 8$	12



$d = \lfloor \log 8 \rfloor = 3$

Αλγόριθμος κ.σ. (κατασκευής σωρού)

ΕΙΣΟΔΟΣ: Δομή σωρού L με n κόμβους

ΕΞΟΔΟΣ: Ίδιο δέντρο, τώρα σωρός

$d = \lfloor \log n \rfloor$
$m_1 = 2^d$
for $m = (d\text{-}1)$ downto 0 do
    $k = m$
    while $2k \leq n$ and $k < m_1$ do
        αλγόριθμος σ.σ (k,n);
        $k = k+1$
    end while
    $m_1 = 2^m$
end m-loop

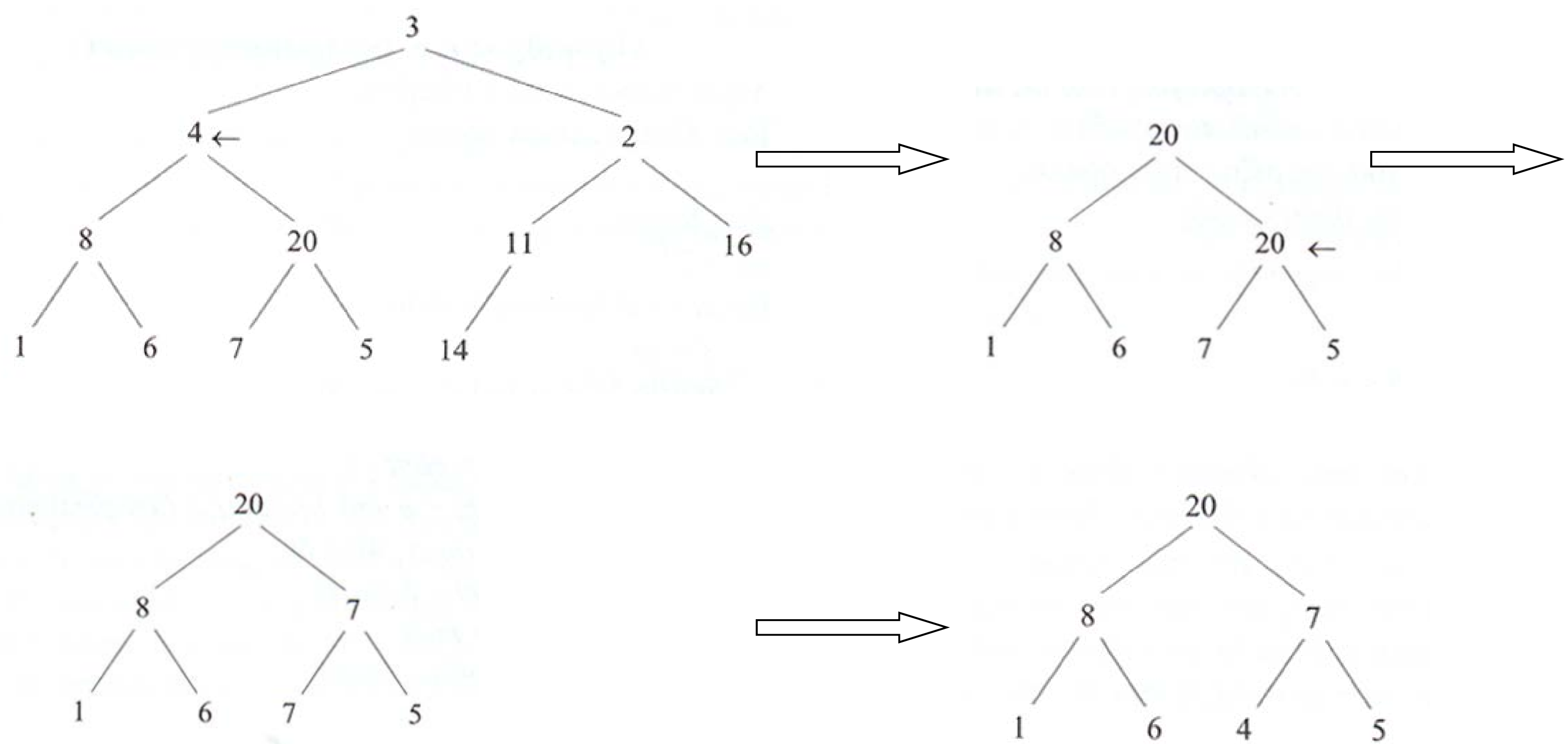
Αλγόριθμος σ.σ. (k, n)

ΕΙΣΟΔΟΣ: Δομή σωρού L με δείκτη ρίζας τον r, με n στοιχεία, όπου κάθε ένα από τα υποδέντρα της ρίζας είναι σωρός

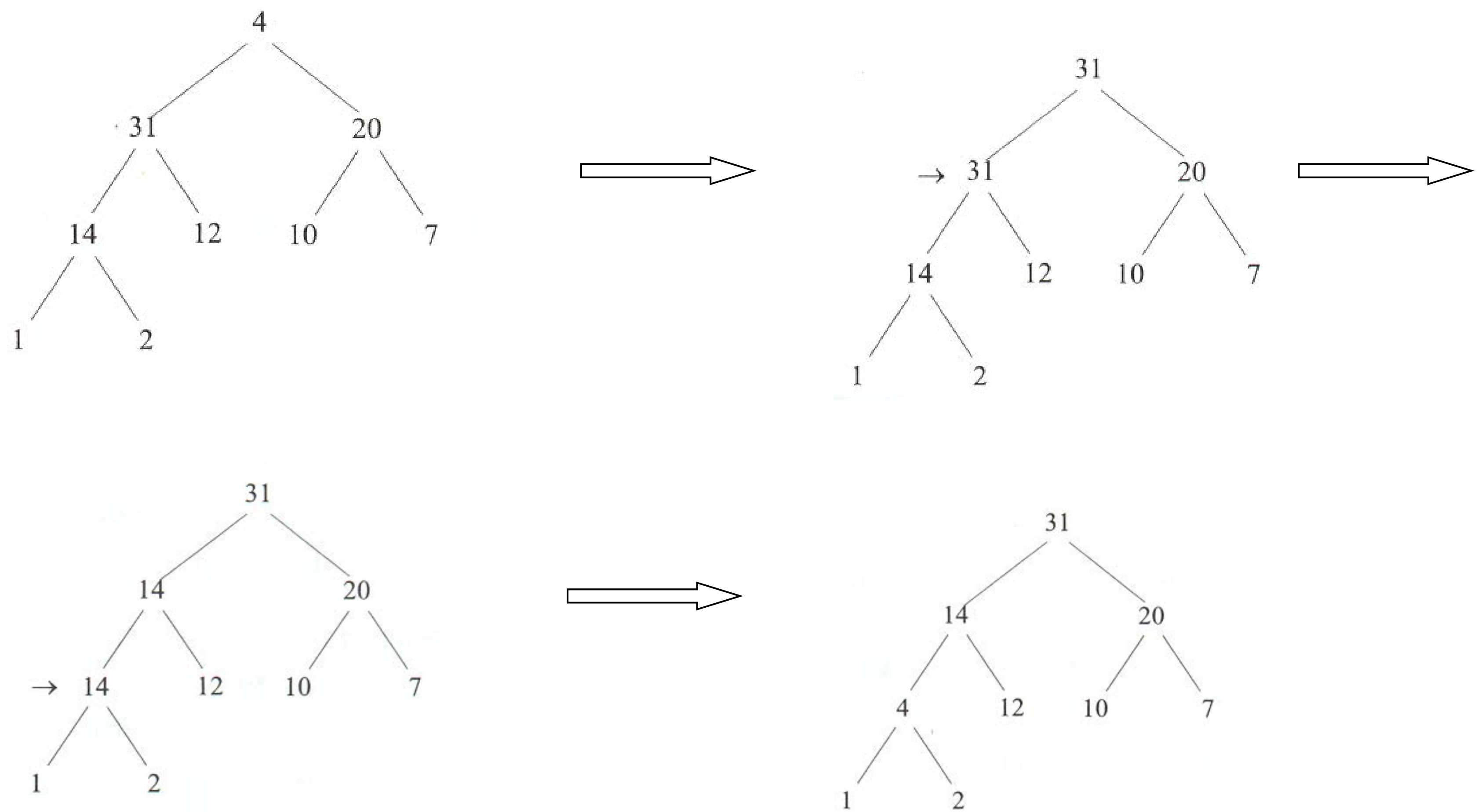
ΕΞΟΔΟΣ: Ίδιο δέντρο, τώρα σωρός

$x = L(r)$
$rleft = 2r$
while $rleft \leq n$ do
    $max = rleft$
    $rright = rleft + 1$
    if $rright \leq n$ and $L(right) \geq L(rleft)$ then $max = rright$;
    if $x < L(max)$ then do
        $L(r) = L(max)$
        $r = max$
        $rleft \leftarrow 2max$
    end if
end while
$L(r) = x$

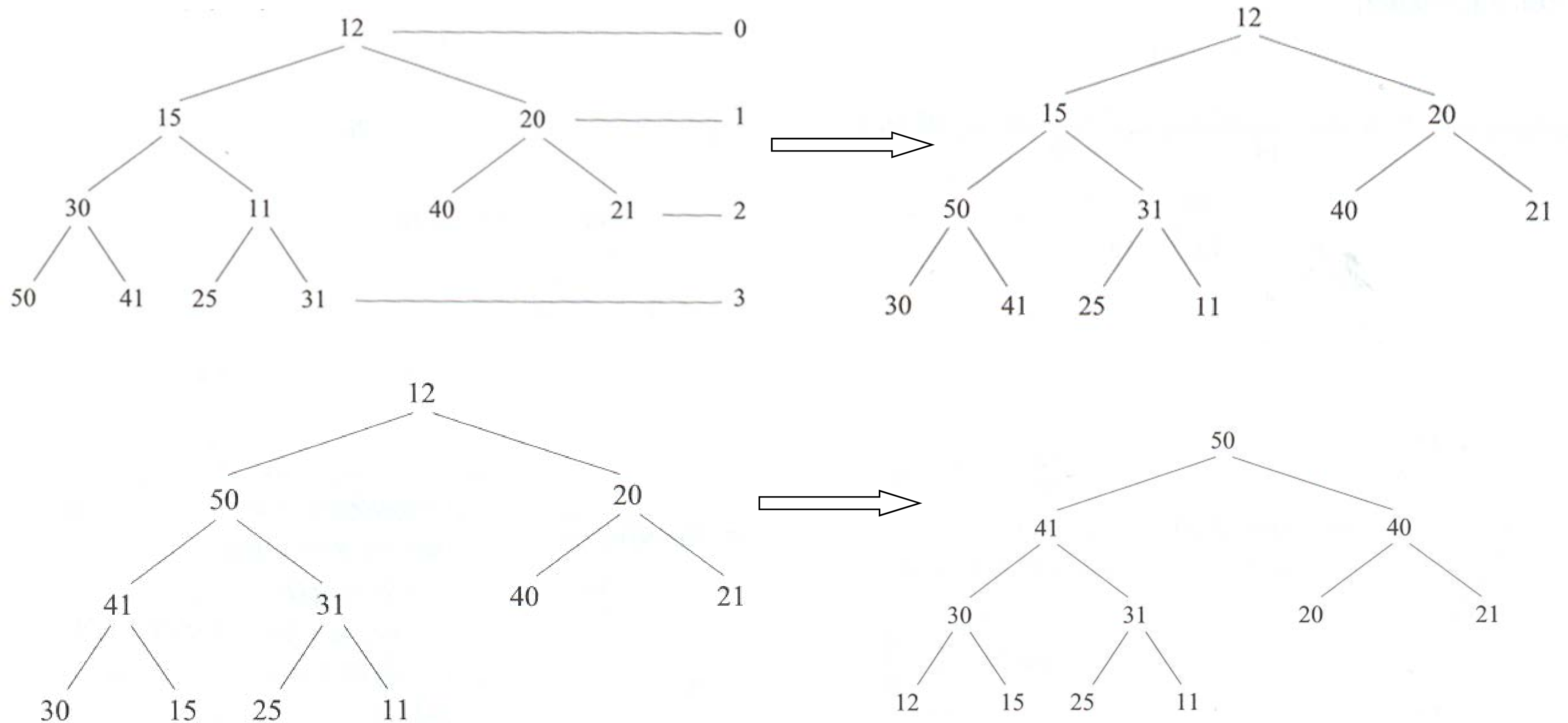
ΠΑΡΑΔΕΙΓΜΑ Π1.12-1. Αλγόριθμος σ.σ. (συμπλήρωσης σωρού)



ΠΑΡΑΔΕΙΓΜΑ Π1.12-2. Να ακολουθηθεί ο *σ.σ.(1,9)* για το παρακάτω δέντρο, αν η είσοδος είναι νόμιμη.



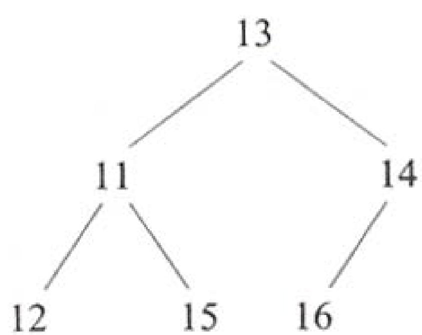
ΠΑΡΑΔΕΙΓΜΑ Π1.12-3. Να ακολουθηθεί ο αλγόριθμος *κ.σ. (κατασκευής σωρού)* για το δέντρο - δομή σωρού.



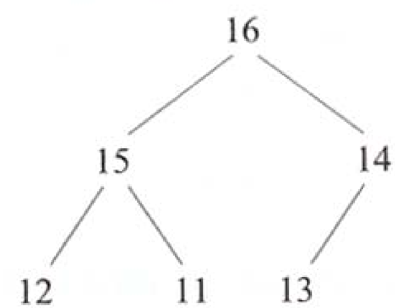
Αλγόριθμος heapsort

αλγόριθμος κ.σ.
for $i = n$ to 2 do
    $\alpha = L(i)$
    αλγόριθμος σ.σ.(1, i-1)
    $L(i) = \alpha$
end

ΠΑΡΑΔΕΙΓΜΑ Π1.12-4. Να ακολουθηθεί ο *Heapsort* για $L=[13,11,14,12,15,16]^T$.
Ισχύει $n=6$ και η αντίστοιχη δομή σωρού είναι:



Μετά από τον κ.σ. προκύπτει:

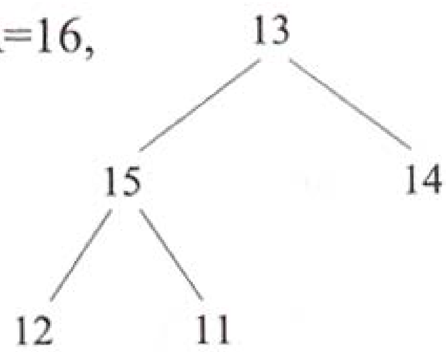


ή

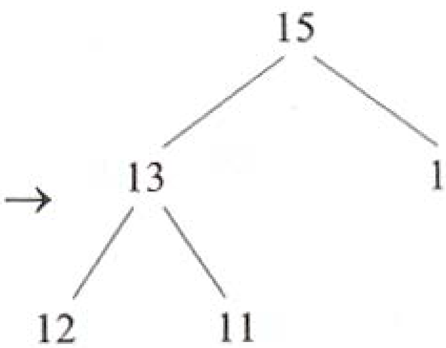
$$L = \begin{bmatrix} 16 \\ 15 \\ 14 \\ 12 \\ 11 \\ 13 \end{bmatrix}$$

Βρίσκουμε διαδοχικά:

i=6 A=16,



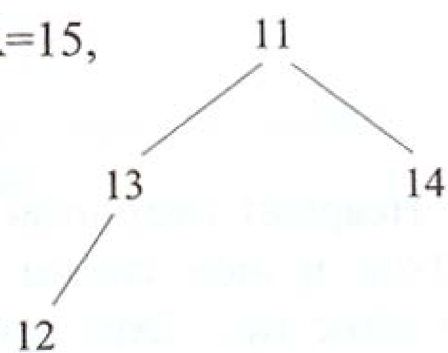
σ.σ. (1,5) →



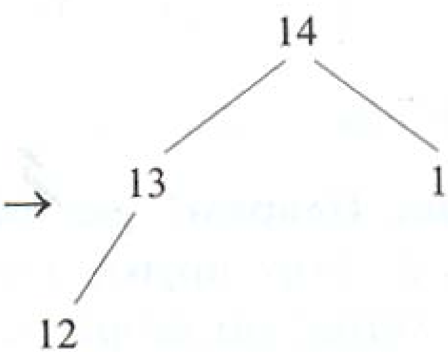
,

$$L=\begin{bmatrix}15\\13\\14\\12\\11\\16\end{bmatrix}$$

i=5 A=15,



σ.σ. (1,4) →



,

$$L=\begin{bmatrix}14\\13\\11\\12\\15\\16\end{bmatrix}$$

.

.

.

(κ.λ.π.)

Υπολογισμός της ΠΧΠ και το βέλτιστο του αλγορίθμου.

Για τις κλίσεις του σ.σ. από τον κ.σ. και για το επίπεδο *j* έχουμε

2* (βάθος <<δέντρου-εισόδου>>)*(αριθμός εσωτ. κόμβων) = 2 (d-j)2^j συγκρίσεις

$$\sigma_1 = \sum_{i=0}^{d-1} 2(d-i)2^i = 2^{d+2} - 2d - 4 < 4n - 2\lfloor \log n \rfloor - 4 \in \Theta(n)$$

Θυμηθείτε ότι γενικά:

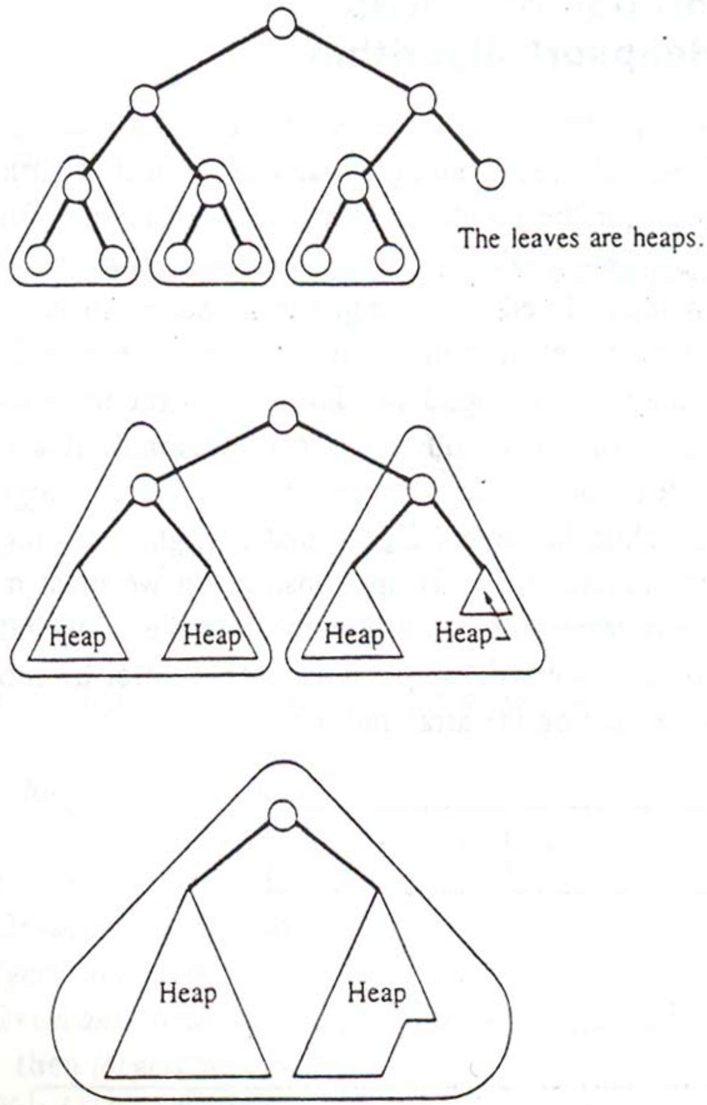
$$\sum_{i=1}^{n} ic^i = c + 2c^2 + \cdots + nc^n = \frac{-(n+1)c^{n+1} + nc^{n+2} + c}{(c-1)^2}$$

Για να βρούμε την *ΠΧΠ* πρέπει να προσθέσουμε και τις σ_2 συγκρίσεις από τις

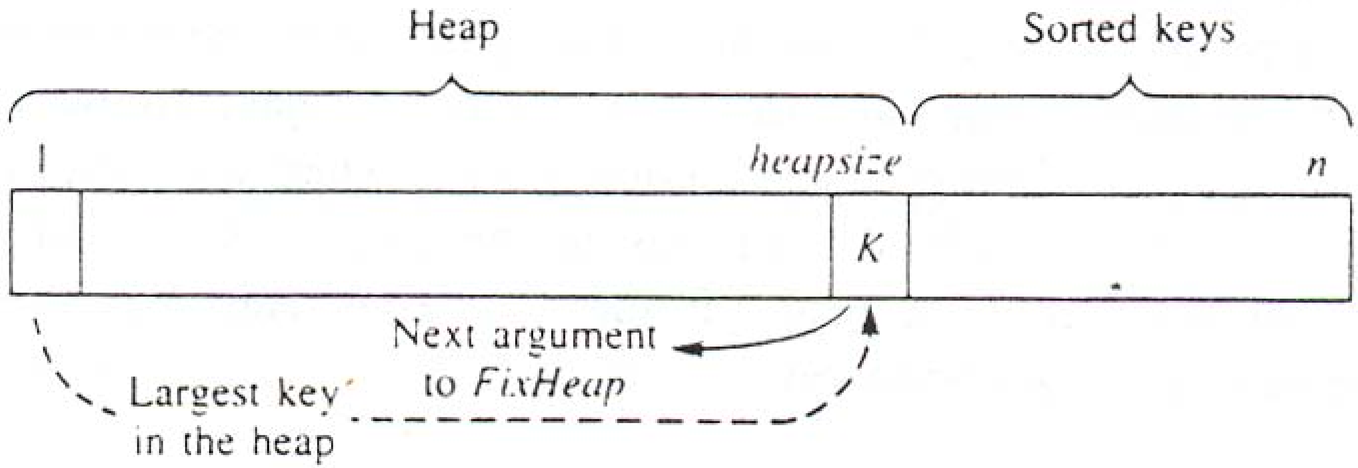
κλίσεις του σ.σ. που γίνονται απευθειας από τον *heapsort* και είναι

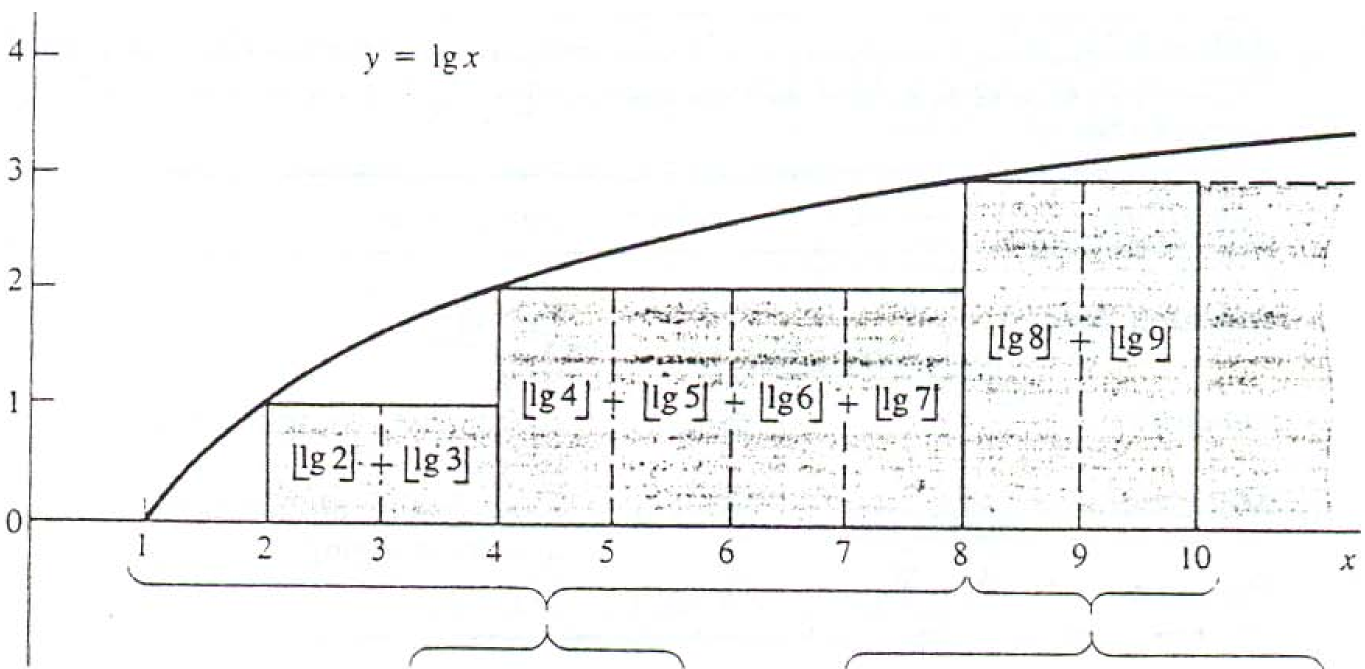
$$\sigma_2 = 2\sum_{k=1}^{n-1} \lfloor \log k \rfloor = \sum_{k=1}^{\lfloor \log n \rfloor - 1} k2^k + \lfloor \log n \rfloor (n - 2^{\lfloor \log n \rfloor})$$

$$\Pi X \Pi(n) = \sigma_1 + \sigma_2 = \Theta(n \log n)$$



The leaves are heaps.
Heap
Heap
Heap
Heap
Heap
Heap





$$\sum_{i=1}^{n-1} \lfloor \lg i \rfloor = \sum_{j=1}^{\lfloor \lg n \rfloor - 1} j 2^j + \lfloor \lg n \rfloor (n - 2^{\lfloor \lg n \rfloor})$$

$n = 10$