Αλγόριθμοι και Πολυπλοκότητα

Ν. Μ. Μισυρλής

Τμήμα Πληροφορικής και Τηλεπικοινωνιών,
Πανεπιστήμιο Αθηνών

Σωροί

Δένδρα

- Ένα δένδρο είναι μια δομή δεδομένων η οποία μπορεί να είναι είτε ένα ατομικό δένδρο (ένα φύλλο), είτε ένας κόμβος και μια ακολουθία από υποδένδρα.
- Οι κόμβοι ενός δένδρου διακρίνονται σε *εσωτερικούς* κόμβους και *φύλλα*. Τα φύλλα είναι κόμβοι οι οποίοι δεν έχουν παιδιά.
- Το βάθος ενός κόμβου είναι το μήκος του μονοπατιού το οποίο ενώνει τον κόμβο αυτό με τη ρίζα. Κατά σύμβαση θεωρούμε πάντα ότι η ρίζα έχει βάθος 0.
- Το ύψος ενός κόμβου είναι το μήκος του μεγαλύτερου μονοπατιού που ενώνει τον κόμβο με τα φύλλα. Το ύψος του δένδρου είναι το ύψος της ρίζας.

Σωροί

Δυαδικά Δένδρα

- Μια ειδική κατηγορία δένδρων είναι τα *δυαδικά δένδρα* στα οποία κάθε κόμβος έχει το πολύ δύο παιδιά.
- Γενικά το επίπεδο i περιέχει το πολύ 2^i κόμβους.
- *Σχεδόν πλήρες* καλείται ένα δυαδικό δένδρο όταν όλες οι γραμμές, εκτός ίσως από την τελευταία, περιέχουν το μέγιστο αριθμό κόμβων (δηλαδή 2^i). Επιπλέον ισχύει μια σειρά από ιδιότητες όπως:
- Τα φύλλα της τελευταίας γραμμής είναι όλα αριστερά
- Τα φύλλα βρίσκονται όλα στην τελευταία και ενδεχομένως στην προτελευταία γραμμή
- Οι εσωτερικοί κόμβοι είναι όλοι δυαδικοί, εκτός από το δεξιότερο της προτελευταίας γραμμής, ο οποίος μπορεί να μην έχει δεξιό παιδί.

Σωροί

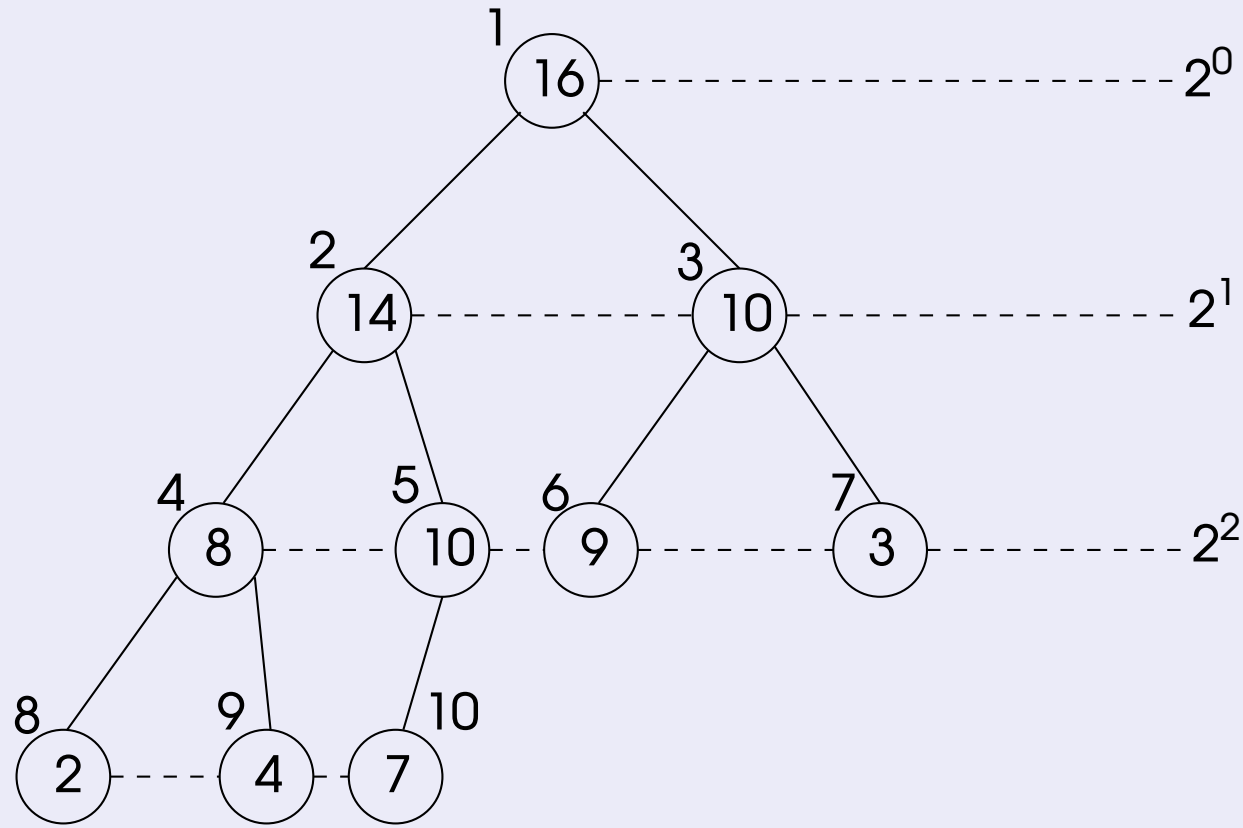
Αρίθμηση κόμβων

- Κάθε κόμβος έχει τον πατέρα του στη θέση $\lfloor i/2 \rfloor$, το αριστερό παιδί του κόμβου i, είναι ο κόμβος $2i$ και το δεξιό παιδί του κόμβου i είναι το $2i + 1$.
- Σε ένα δυαδικό δένδρο με m κόμβους και ύψος h ισχύει

$$\lfloor \log_2 m \rfloor \leq h \leq m - 1$$

Σωροί

Σχεδόν πλήρες δυαδικό δένδρο



Σωροί

Η δομή σωρού

- Έστω ότι πελάτες παρουσιάζονται στο ταμείο μιας τράπεζας με ένα νούμερο σε ένα χαρτί αντιπροσωπεύοντας τον αριθμό προτεραιότητας του καθενός.Χρειαζόμαστε:
- Αναζήτηση του μέγιστου αριθμού στην ουρά
- Διαγραφή αυτού του στοιχείου από την ουρά
- Εισαγωγή ενός νέου στοιχείου στην ουρά
- Μια πιθανή λύση θα ήταν η ταξινόμηση κατά αύξουσα σειρά των στοιχείων της ουράς. Έτσι η διαγραφή και η αναζήτηση του μεγίστου γίνεται σε σταθερό χρόνο , αλλά η εισαγωγή απαιτεί γραμμικό χρόνο.

Σωροί

Η δομή σωρού (συν.)

- Μια άλλη λύση θα ήταν απλά μια ουρά, αλλά τότε ενώ η εισαγωγή θα γινόταν πολύ γρήγορα, η διαγραφή και η αναζήτηση στην ουρά θα χρειάζονταν επίσης γραμμικό χρόνο.
- Η ουρά n στοιχείων παριστάνεται από ένα δυαδικό δένδρο το οποίο σε κάθε κόμβο περιέχει ένα στοιχείο της ουράς. Το δένδρο αυτό θα πληρεί δύο βασικές ιδιότητες:
- Η τιμή κάθε κόμβου είναι μεγαλύτερη ή ίση της τιμής των παιδιών του κόμβου ιδιότητα σωρού
- Το δέντρο είναι σχεδόν πλήρες δομική ιδιότητα
- Η αναπαράσταση με σωρό μας επιτρέπει να κάνουμε τις πράξεις της αναζήτησης του μέγιστου στοιχείου, της εισαγωγής και της διαγραφής σε χρόνο $O(\log n)$.

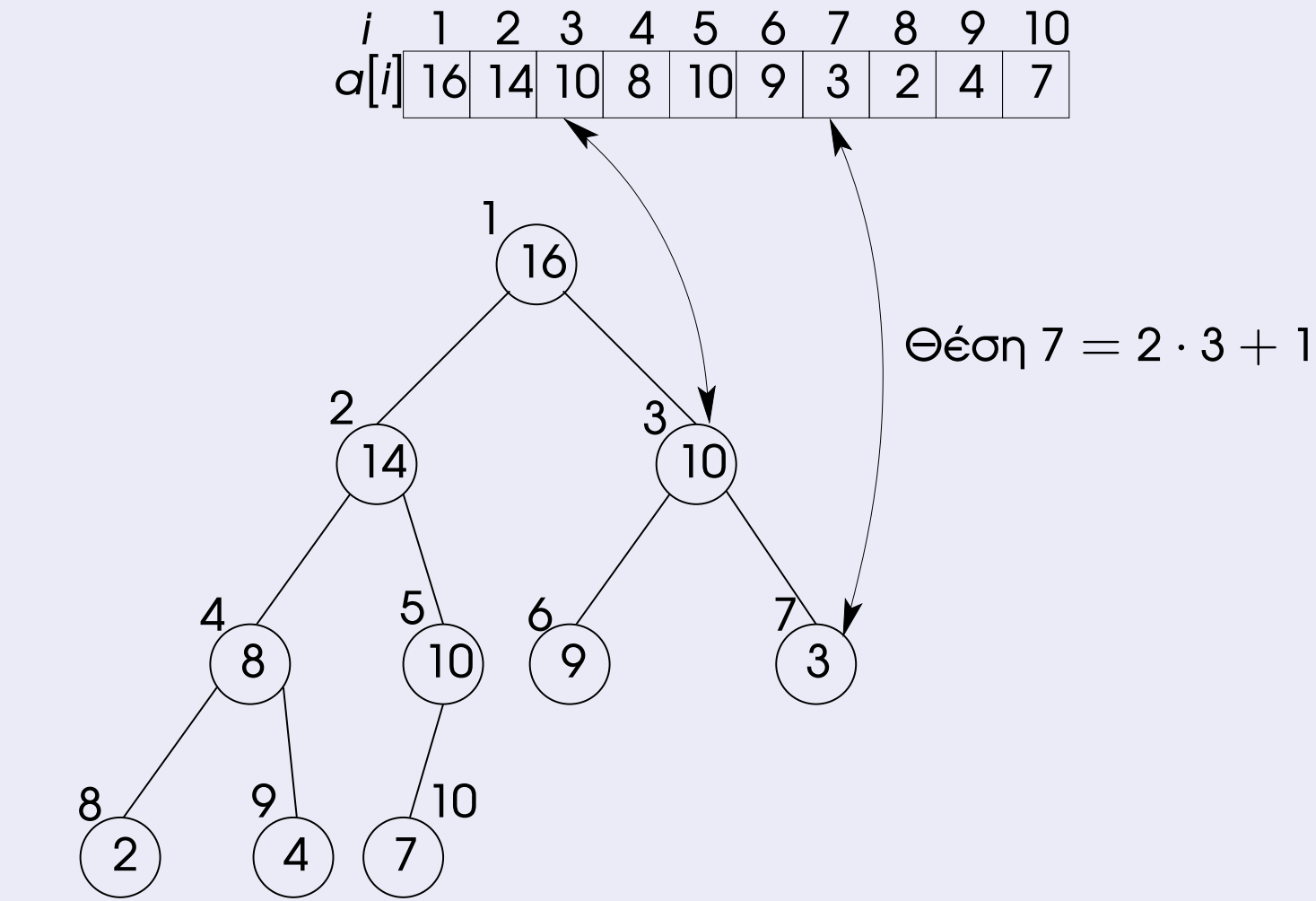
Σωροί

Υλοποίηση με πίνακα ενός σωρού

- Κατά πλάτος αρίθμηση των κόμβων του δέδρου
- Το νούμερο κάθε κόμβου του δένδρου δίνει τον δείκτη του πίνακα που περιέχει την τιμή του κόμβου.

Σωροί

Ένας σωρός και η υλοποίησή του με πίνακα



Σωροί

Εισαγωγή

- Εισάγουμε το στοιχείο σαν τελευταίο φύλλο στο πλήρες δυαδικό δένδρο. Συγκρίνουμε την τιμή του με αυτήν του πατέρα του και αν έχει μεγαλύτερη τιμή, τις αντιμεταθέτουμε. Η διαδικασία επαναλαμβάνεται εωσότου ικανοποιείται η συνθήκη του σωρού.
- Στην χείριστη περίπτωση η πολυπλοκότητα είναι $O(\log n)$.

Σωροί

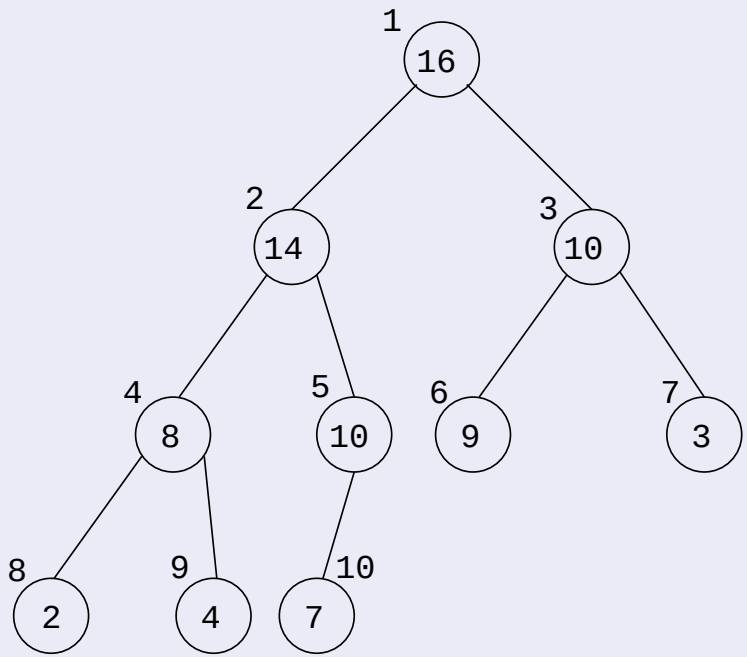
Αλγόριθμος εισαγωγής

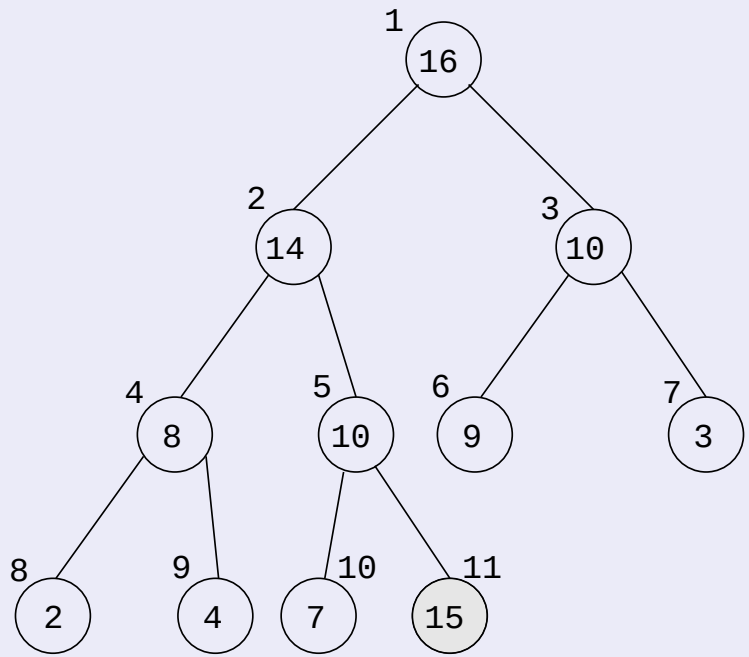
ΣΩΡΟΣ-ΕΙΣΑΓΩΓΗ $(u$: στοιχείο$)$

1. $n = n + 1$
2. $k = n,\ a[k] = u$
3. **while** $a[k/2] < a[k]$
4. $\quad swap\ (a[k],\ a[k/2])$
5. $\quad k = \lfloor k/2 \rfloor$
6. **end while**

Σωροί

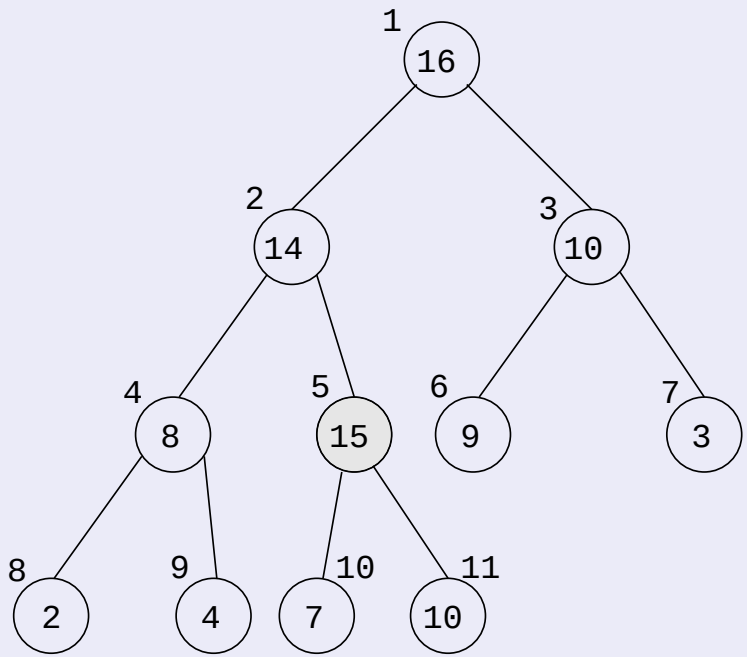
Εισαγωγή ενός νέου στοιχείου στο σωρό(1)

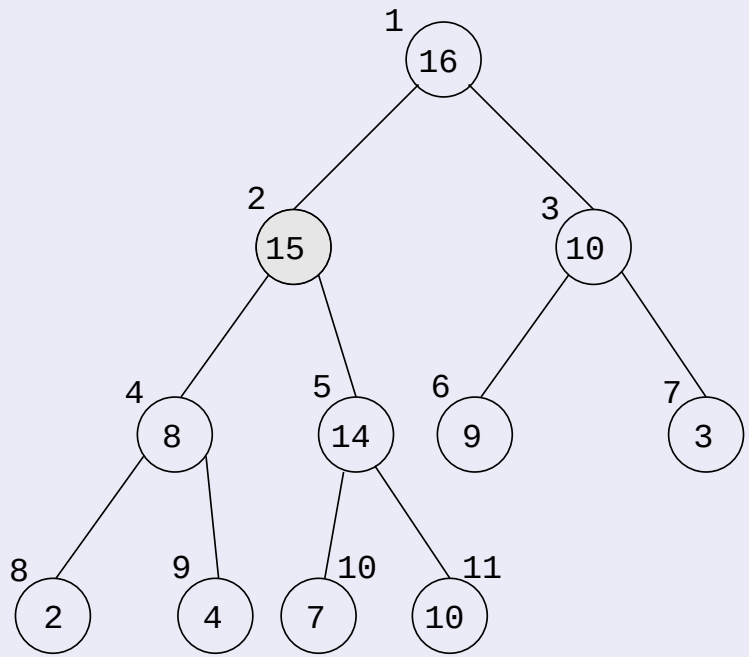




Σωροί

Εισαγωγή ενός νέου στοιχείου στο σωρό(2)





Σωροί

Διαγραφή

- Η διαδικασία διαγραφής του 1ου στοιχείου του σωρού αντικαθιστά τη ρίζα του δέντρου που παριστάνει τον σωρό, με το δεξιότερο φύλλο του τελευταίου επιπέδου του δένδρου.
- Έπειτα συγκρίνεται η τιμή της νέας ρίζας με των παιδιών της και αν είναι μικρότερη από κάποια, την αντιμεταθέτουμε με την μεγαλύτερη.
- Η διαδικασία επαναλαμβάνεται για το νέο στοιχείο, μέχρι να ικανοποιηθεί η συνθήκη του σωρού.
- Στην χείριστη περίπτωση η πολυπλοκότητα είναι $O(\log n)$.

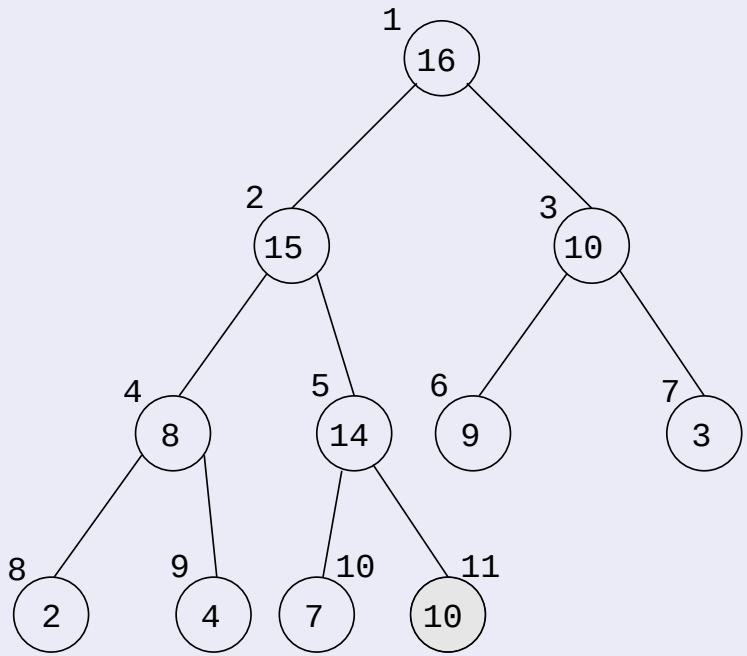
Σωροί

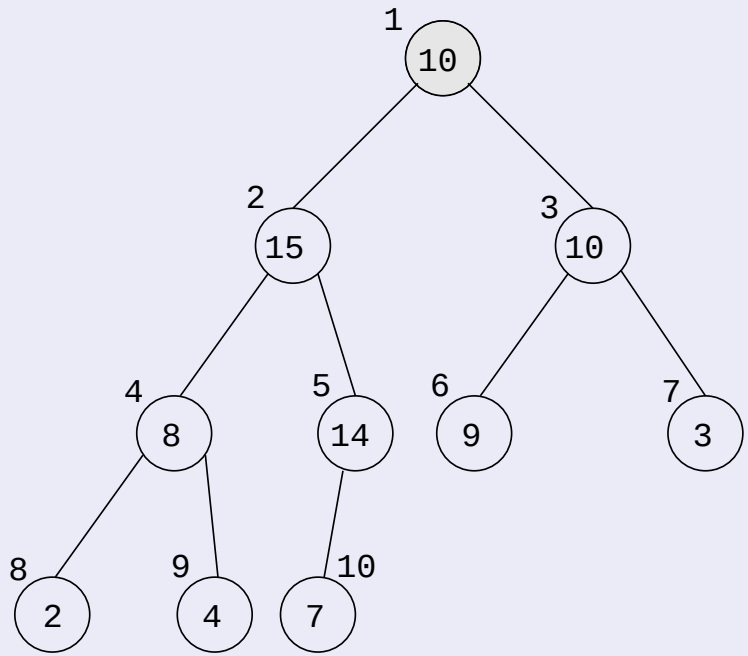
Σωρός -Διαγραφή

1. $a[1] = a[n]$
2. $n = n - 1, i = 1$
3. **do**
4. $\quad l = 2 \cdot i, r = 2 \cdot i + 1$
5. $\quad$ **if** $l \leq n$ **and** $a[l] > a[i]$ **then** $max = l$
6. $\quad$ **else** $max = i$
7. $\quad$ **if** $r \leq n$ **and** $a[r] > a[max]$ **then** $max = r$
8. $\quad$ **if** $i \neq max$ **then** $swap\ (a[i], a[max]), i = max$
9. $\quad$ **else** break
10. **while** $(i < n)$

Σωροί

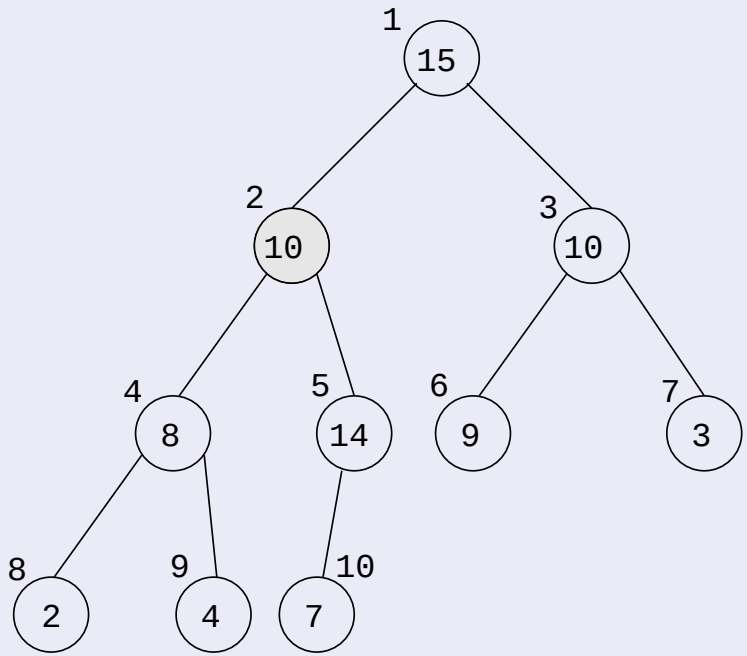
Διαγραφή της ρίζας από το το σωρό (1)

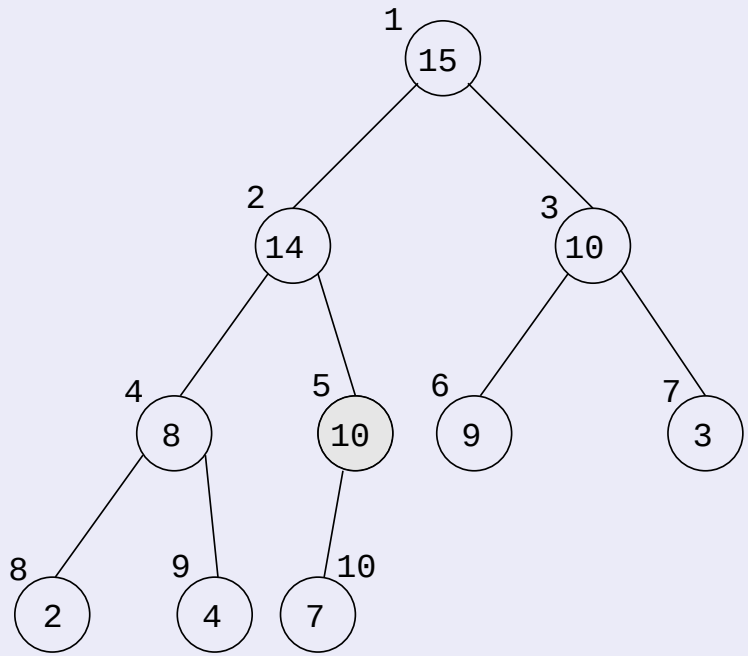




Σωροί

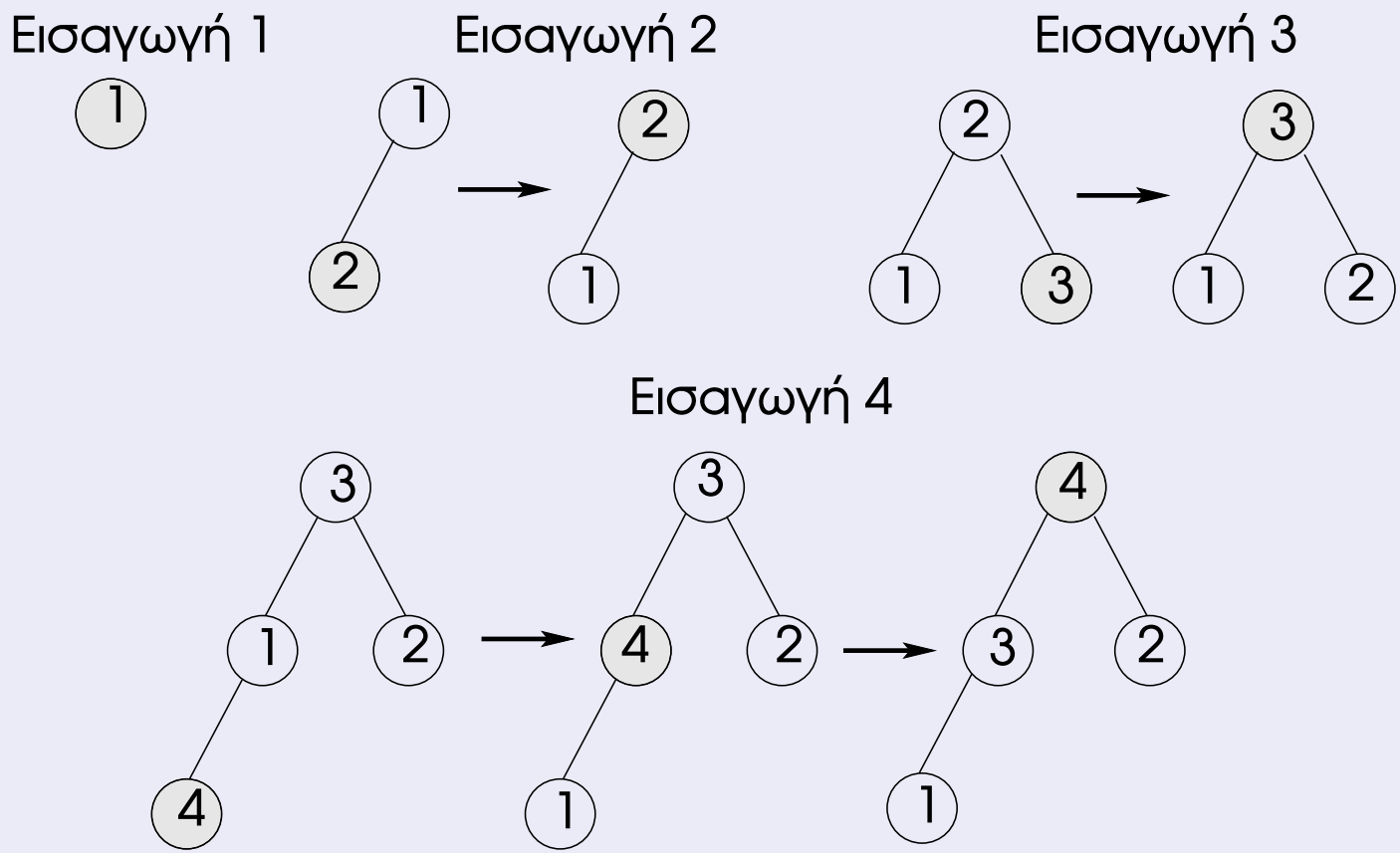
Διαγραφή της ρίζας από το το σωρό (2)





Σωροί

Εισαγωγή ενός νέου στοιχείου στο σωρό



Σωροί

Αλγόριθμος κατασκευής σωρού

ΚΑΤΑΣΚΕΥΗ-ΣΩΡΟΥ (πίνακας a με n στοιχεία)

1. **for** $i = 1$ **to** n **do**
2. ΣΩΡΟΣ-ΕΙΣΑΓΩΓΗ $(a[i])$
3. **end for**

Σωροί

Αλγόριθμος διατήρησης ιδιότητας σωρού

ΔΙΑΤΗΡΗΣΗ-ΣΩΡΟΥ(πίνακας a με n στοιχεία, θέση i)

1. $l = 2 \cdot i$
2. $r = 2 \cdot i + 1$
3. **if** $l \leq n$ **and** $a[l] > a[i]$
4. $\quad max = l$
5. **else**
6. $\quad max = i$
7. **if** $r \leq n$ **and** $a[r] > a[max]$
8. $\quad max = r$
9. **if** $i \neq max$ **then**
10. $\quad swap\ (a[i],\ a[max])$
11. $\quad$ ΔΙΑΤΗΡΗΣΗ-ΣΩΡΟΥ (a, max)

Σωροί

Αλγόριθμος κατασκευής σωρού σε γραμμικό χρόνο

- Θεωρούμε τώρα ότι στην αταξινόμητη ακολουθία όλα τα στοιχεία που είναι φύλλα του σωρού είναι υπο-σωροί που πληρούν την ιδιότητα του σωρού.
- Για κάθε επόμενο εσωτερικό κόμβο (από κάτω προς τα πάνω, δηλαδή από $\lfloor n/2 \rfloor$ εώς 1) επιβάλλουμε την συνέπεια με την ιδιότητα σωρού χρησιμοποιώντας τον αλγ·οριυμο ΔΙΑΤΗΡΗΣΗ-ΣΩΡΟΥ

ΚΑΤΑΣΚΕΥΗ-ΣΩΡΟΥ (πίνακας a με n στοιχεία)

1. **for** $i = \lfloor n/2 \rfloor$ **to** 1 **do**
2. ΔΙΑΤΗΡΗΣΗ-ΣΩΡΟΥ (a, i)
3. **end for**

Σωροί

Κατασκευή Σωρού

- Υποθέτουμε ότι τα στοιχεία έρχονται με μια σειρά και δεν είναι γνωστά εκ των προτέρων.
  Στην χείριστη περίπτωση (αύξουσα ακολουθία), αν εισάγεται στο βήμα i, τότε θα χρειαστεί $O(\log i)$ βήματα.
- Ο χρόνος του αλγορίθμου φράσσεται ως:

$$O(\log 1) + O(\log 2) + \ldots + O(\log n) = O(n \log n)$$

Σωροί

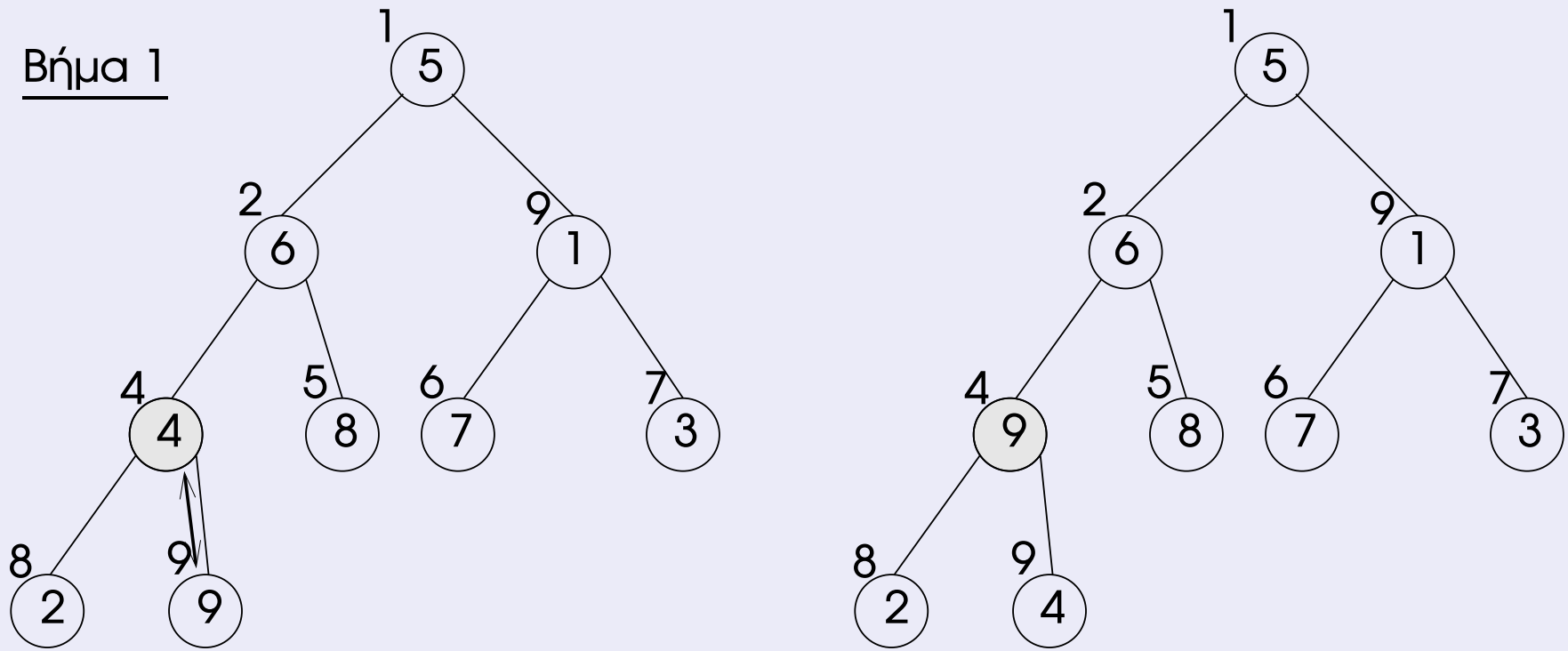
Κατασκευή Σωρού (συν.)

- Μπορούμε να επιτύχουμε καλύτερο χρόνο για την κατασκευή του σωρού αν γνωρίζουμε από την αρχή όλην την ακολουθία εισόδου και την έχουμε αποθηκευμένη σε έναν πίνακα.
- Ο αρχικός πίνακας εισόδου a είναι η αναπαράσταση ενός αταξινόμητου σωρού, επιβάλλουμε την ιδιότητα του σωρού, από τους χαμηλότερους εσωτερικούς κόμβους και πηγαίνοντας προς την ρίζα.
- Ο αλγόριθμος συγκρίνει το τρέχον στοιχείο με τα παιδιά του ανυψώνοντας το μεγαλύτερο, εωσότου και τα δύο παιδιά του έχουν μικρότερη τιμή.

Σωροί

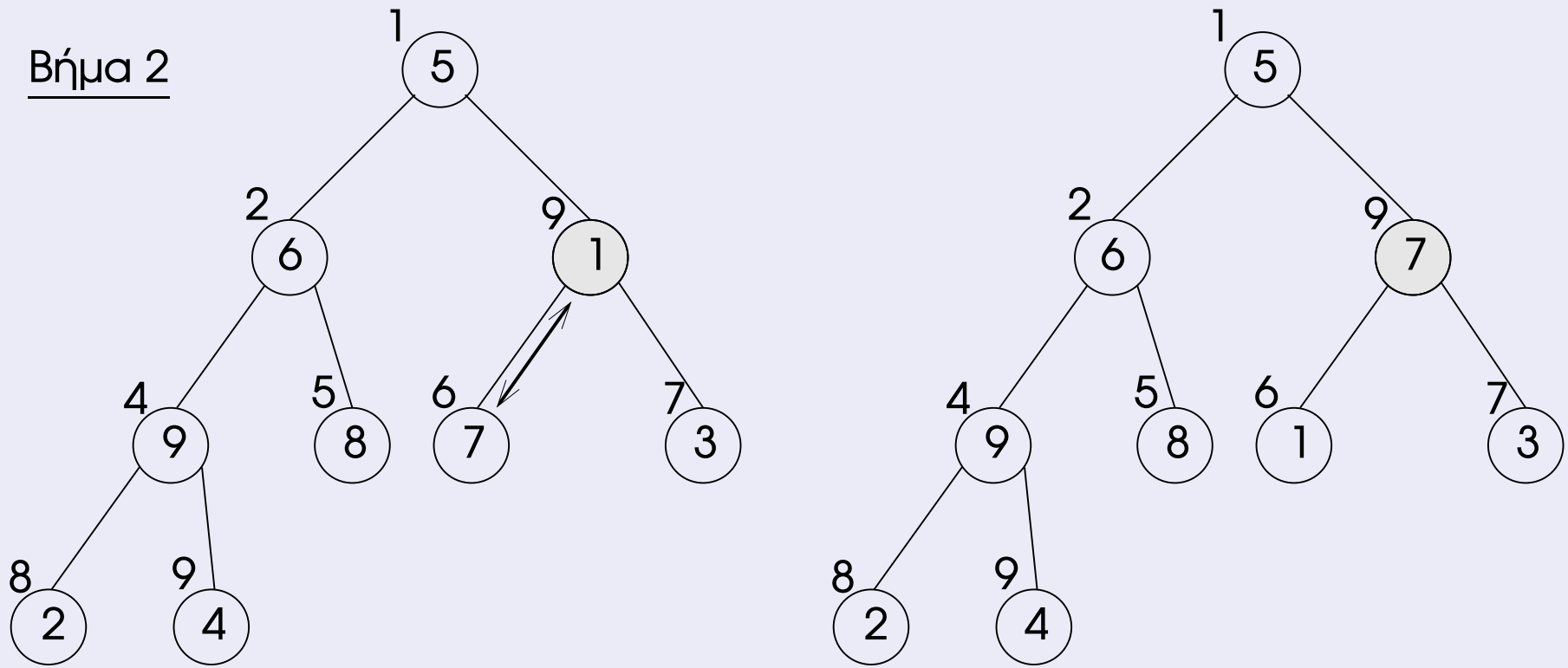
Αλγόριθμος κατασκευής σωρού σε γραμμικό χρόνο(1)

- Εκτέλεση του αλγορίθμου κατασκευής του σωρού, όταν ο πίνακας εισόδου είναι ο $[5, 6, 1, 4, 8, 7, 3, 2, 9]$.



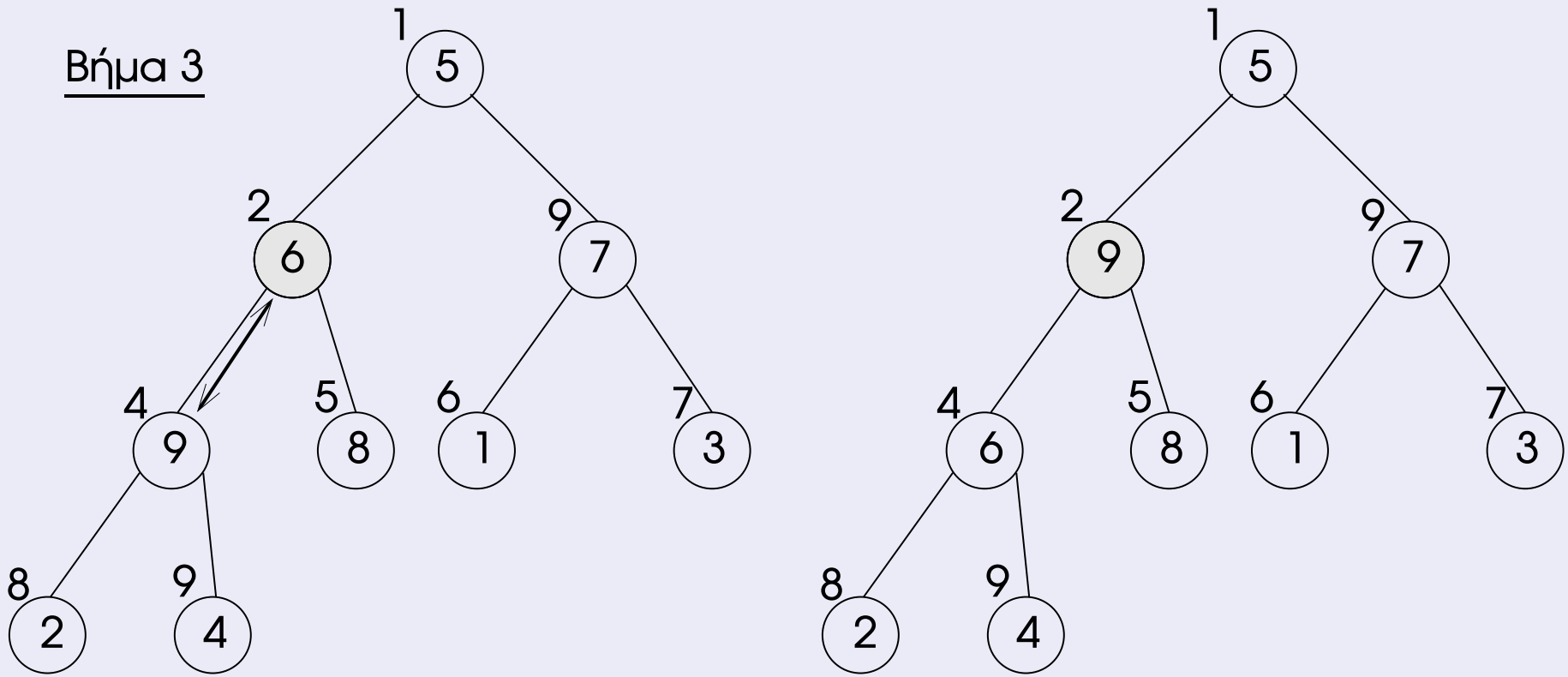
Σωροί

Αλγόριθμος κατασκευής σωρού σε γραμμικό χρόνο (2)



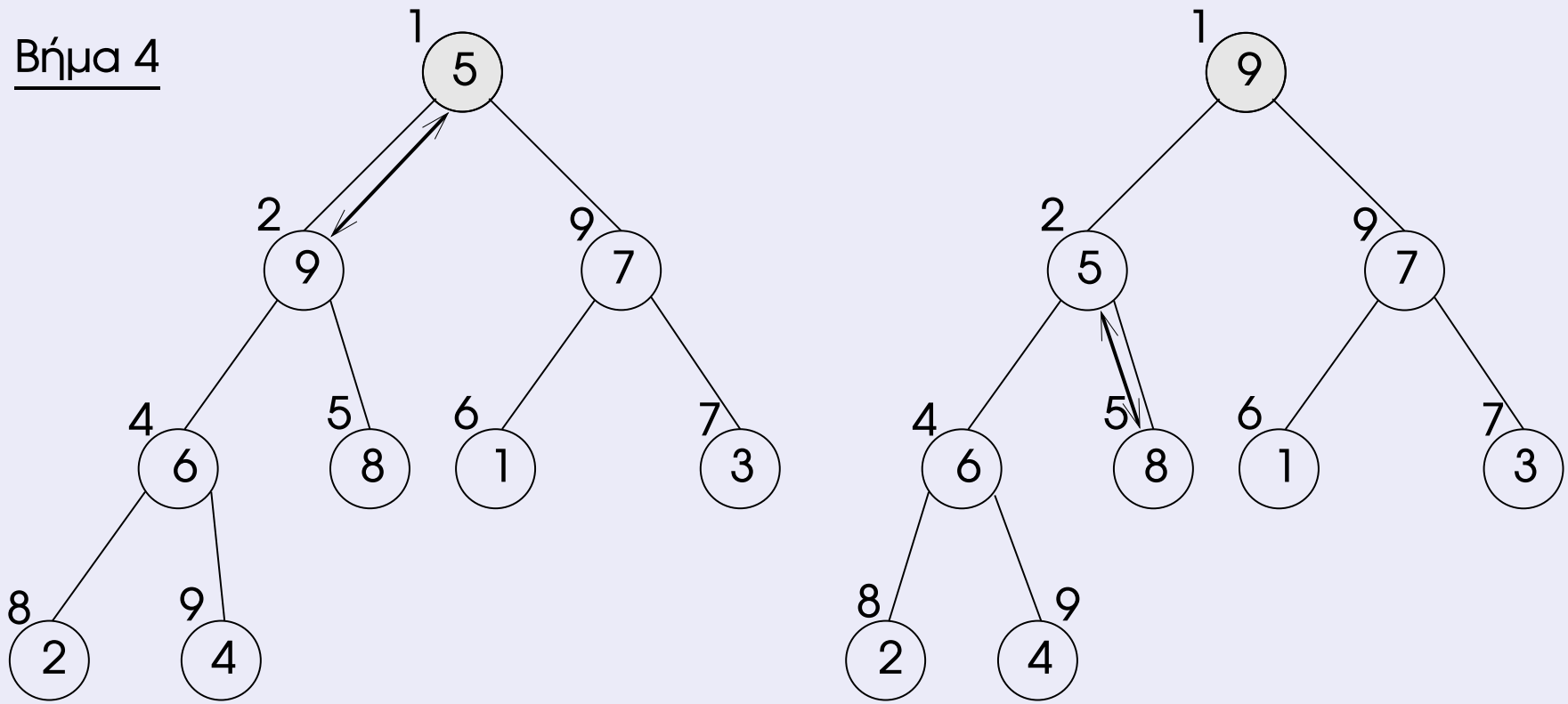
Σωροί

Αλγόριθμος κατασκευής σωρού σε γραμμικό χρόνο (3)



Σωροί

Αλγόριθμος κατασκευής σωρού σε γραμμικό χρόνο (4)



Σωροί

Πολυπλοκότητα

- Η πολυπλοκότητα μίας κλήσης της συνάρτησης ΔΙΑΤΗΡΗΣΗ-ΣΩΡΟΥ είναι $O(h)$ άρα η συνολική πολυπλοκότητα της ΚΑΤΑΣΚΕΥΗ-ΣΩΡΟΥ φράσσεται από

$$\sum_{h=0}^{\lfloor \log n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil O(h) = O\left(n \sum_{h=0}^{\lfloor \log n \rfloor} \frac{h}{2^h}\right)$$

Σωροί

Πολυπλοκότητα (συν.)

Επειδή τώρα

$$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$$

για $x = 1/2$ έχουμε

$$\sum_{k=0}^{\infty} \frac{k}{2^k} = 2$$

- Άρα η πολυπλοκότητα της κατασκευής του σωρού φράσσεται από $O(n \cdot 2) = O(n)$.

Σωροί

Ταξινόμηση με σωρό

- Αρχικά, κατασκευάζουμε τον σωρό των προς ταξινόμηση στοιχείων και κάνουμε διαδοχικές διαγραφές της ρίζας του δένδρου.
- Η συνάρτηση ΜΕΓΙΣΤΟΣ-ΣΩΡΟΥ( ) επιστρέφει την ρίζα του σωρού.

HEAPSORT (πίνακας a με n στοιχεία)

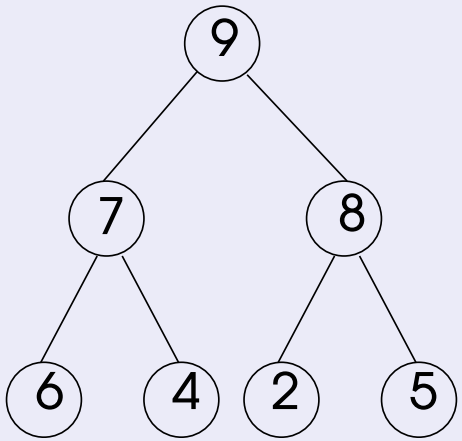
1. ΚΑΤΑΣΚΕΥΗ-ΣΩΡΟΥ (a)
2. **for** $i = 1$ **to** n
3. $\quad a[i] =$ ΜΕΓΙΣΤΟΣ-ΣΩΡΟΥ( )
4. $\quad$ ΣΩΡΟΣ-ΔΙΑΓΡΑΦΗ( )

Σωροί

Ταξινόμηση με σωρό

Παράδειγμα: Να ταξινομηθούν με τη βοήθεια σωρού τα παρακάτω στοιχεία $6, 9, 2, 7, 4, 5, 8$.

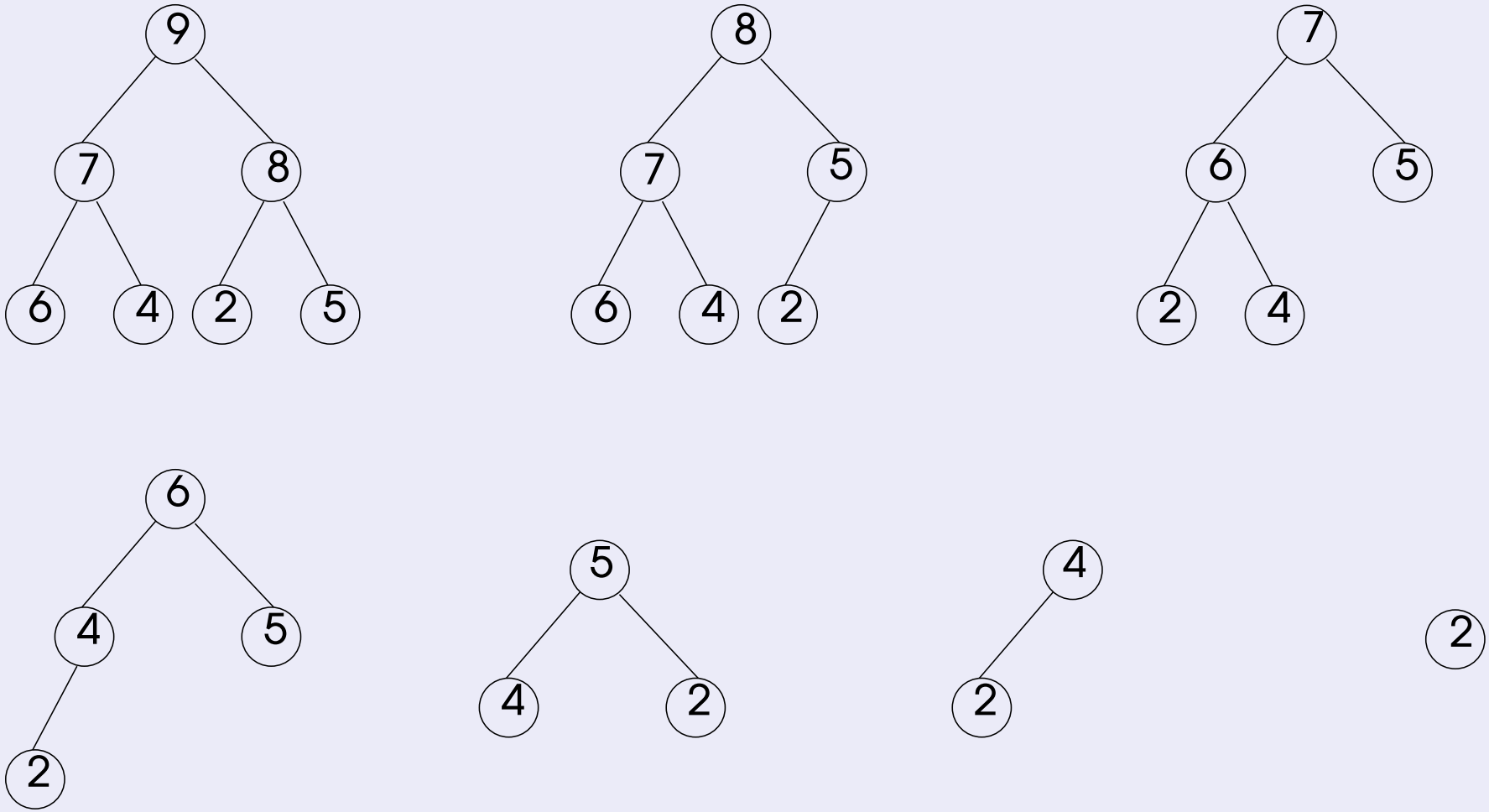
Λύση: Κατασκευάζουμε το σωρό.



Σχήμα: Ο σωρός

Σωροί

Ταξινόμηση με σωρό



Σωροί

Πολυπλοκότητα

- Αφαιρούμε διαδοχικά το μεγαλύτερο στοιχείο.
- Το πρώτο βήμα του αλγορίθμου απαιτεί χρόνο $O(n)$, το δεύτερο απαιτεί χρόνο $O(n \log n)$.
- Συνολικά επομένως ο αλγόριθμος ταξινόμησης με σωρό έχει πολυπλοκότητα $O(n \log n)$.