Δομές Δεδομένων

Ουρές Προτεραιότητας

Δημήτρης Μιχαήλ

Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Ουρά Προτεραιότητας

Το πρόβλημα

Έχουμε αντικείμενα με κλειδιά και θέλουμε ανά πάσα στιγμή να επεξεργαζόμαστε το αντικείμενο με το μικρότερο κλειδί. Στη συνέχεια θέλουμε να ανανεώσουμε την συλλογή αντικειμένων και να επεξεργαστούμε εκ νέου το αντικείμενο με το μικρότερο κλειδί.

Ουρά Προτεραιότητας

Ορισμός

Μία **ουρά προτεραιότητας** (priority queue) είναι μία δομή δεδομένων στοιχείων με κλειδιά η οποία υποστηρίζει τρεις βασικές λειτουργίες:

- εισαγωγή ενός νέου στοιχείου
- εύρεση του στοιχείου με το μικρότερο κλειδί
- διαγραφή του στοιχείου με το μικρότερο κλειδί

Παραδείγματα Εφαρμογών

- σύστημα προσομοίωσης, όπου τα κλειδιά αντιστοιχούν σε χρόνους συμβάντων
- χρονοπρογραμματισμό εργασιών όπου τα κλειδιά είναι προτεραιότητες
- ταξινόμηση αριθμών (θα αναπτύξουμε αυτό το θέμα σε λεπτομέρεια αργότερα)

και άλλα που θα δούμε σε άλλα μαθήματα

- αλγόριθμος A^*
- αλγόριθμος Dijkstra
- κωδικοποίηση Huffman
- αλγόριθμος Prim - εύρεση ελάχιστων γεννητικών δέντρων (MST)

Διάφορες Λειτουργίες

Μία ουρά προτεραιότητας υποστηρίζει συνήθως:

- κατασκευή ουράς προτεραιότητας από n δεδομένα στοιχεία
- εισαγωγή νέου στοιχείου
- διαγραφή ελάχιστου
- αλλαγή προτεραιότητας στοιχείου ((μείωση κλειδιού)
- διαγραφή στοιχείου
- ένωση δύο ουρών προτεραιότητας

Αυτές οι λειτουργίες μπορούν να λειτουργήσουν και ως ΑΤΔ.

Στοιχειώδης Υλοποίηση

Με Πίνακα

- εισαγωγή νέου στοιχείου: στο τέλος του πίνακα
- διαγραφή ελαχίστου: γραμμική αναζήτηση ελαχίστου στον πίνακα
- κ.τ.λ

Άλλες απλές υλοποιήσεις είναι με

- διατεταγμένο πίνακα (πίνακα όπου τα στοιχεία είναι ταξινομημένα με βάση τα κλειδιά)
- με λίστες
- με διατεταγμένες λίστες

Χρόνοι Υλοποιήσεων

	εισαγωγή	διαγραφή ελαχίστου	διαγραφή	εύρεση ελαχίστου	αλλαγή προτερ.	ένωση
πίνακας	1	n	1	n	1	n
διατετ. πίνακας	n	1	n	1	n	n
λίστα	1	n	1	n	1	1
διατετ. λίστα	n	1	1	1	n	n

Δυαδικός Σωρός

Ένα δέντρο είναι **διατεταγμένο σε σωρό** (heap-ordered) αν το κλειδί κάθε κόμβου είναι μικρότερο ή ίσο από τα κλειδιά όλων των παιδιών του.

Δυαδικός Σωρός

Ένα δέντρο είναι **διατεταγμένο σε σωρό** (heap-ordered) αν το κλειδί κάθε κόμβου είναι μικρότερο ή ίσο από τα κλειδιά όλων των παιδιών του.

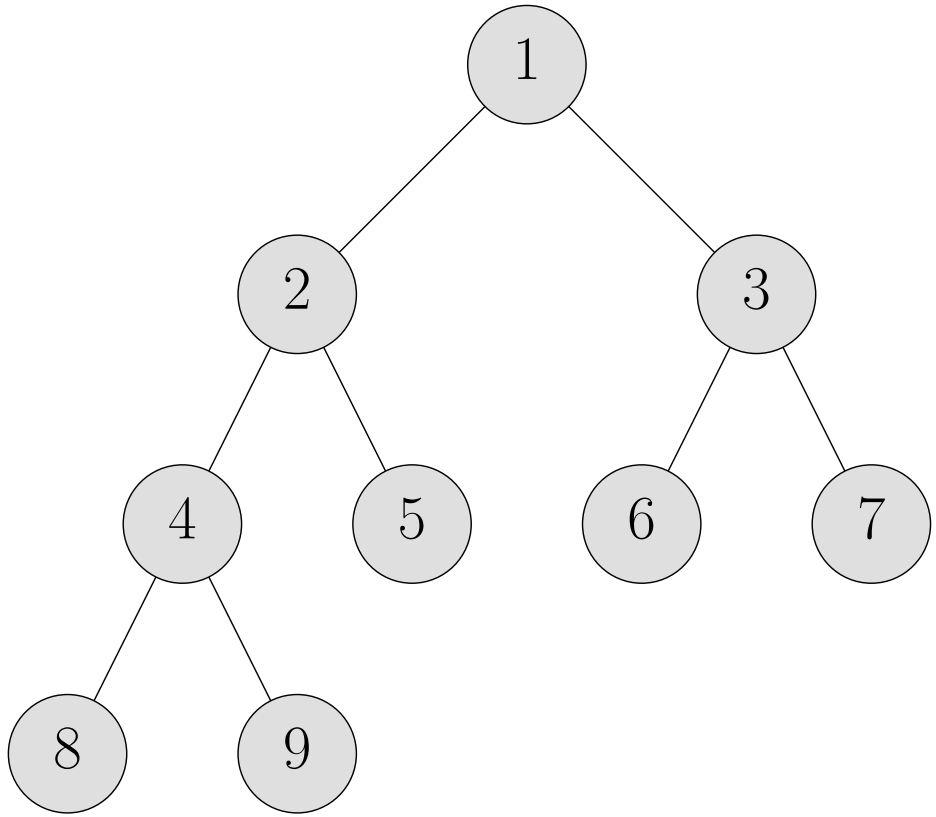
Δυαδικός Σωρός Ελαχίστου (binary min heap)

Ο **σωρός** είναι ένα σύνολο κόμβων με κλειδιά τοποθετημένα σε ένα πλήρες δυαδικό δέντρο το οποίο είναι διατεταγμένο σε σωρό και αναπαριστάται ως πίνακας.

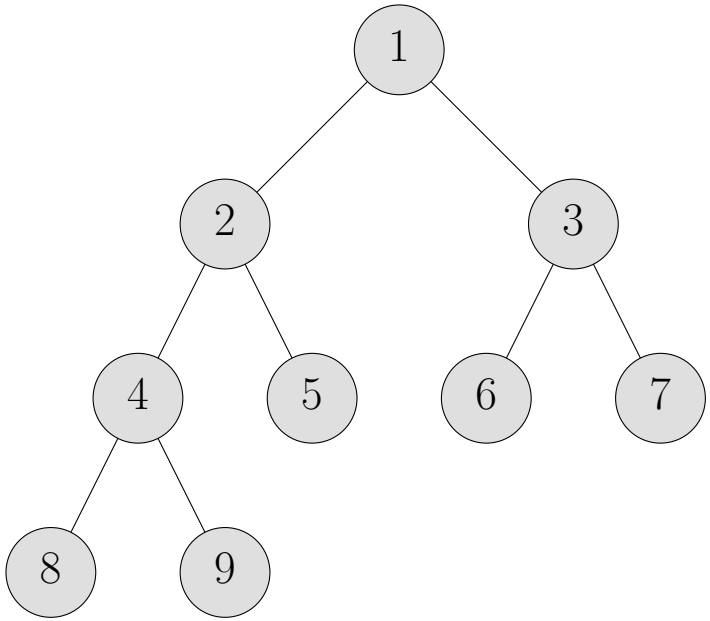
Πλήρες Δυαδικό Δέντρο

Ορισμός

Ένα δυαδικό δέντρο όπου όλα τα επίπεδα, εκτός ίσως του τελευταίου, είναι συμπληρωμένα, το οποίο συμπληρώνεται από τα αριστερά προς τα δεξιά.



Πλήρες Δυαδικό Δέντρο σε Πίνακα

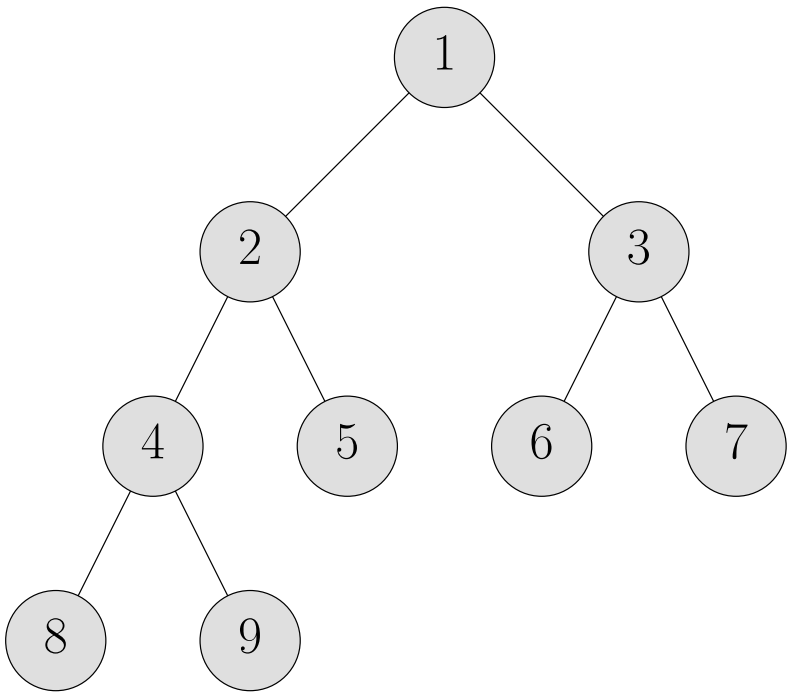


1	2	3	4	5	6	7	8	9

- $\mathit{parent}(i) = \lfloor i/2 \rfloor$
- $\mathit{left-child}(i) = 2i$
- $\mathit{right-child}(i) = 2i + 1$

Ύψος Πλήρους Δυαδικού Δέντρου

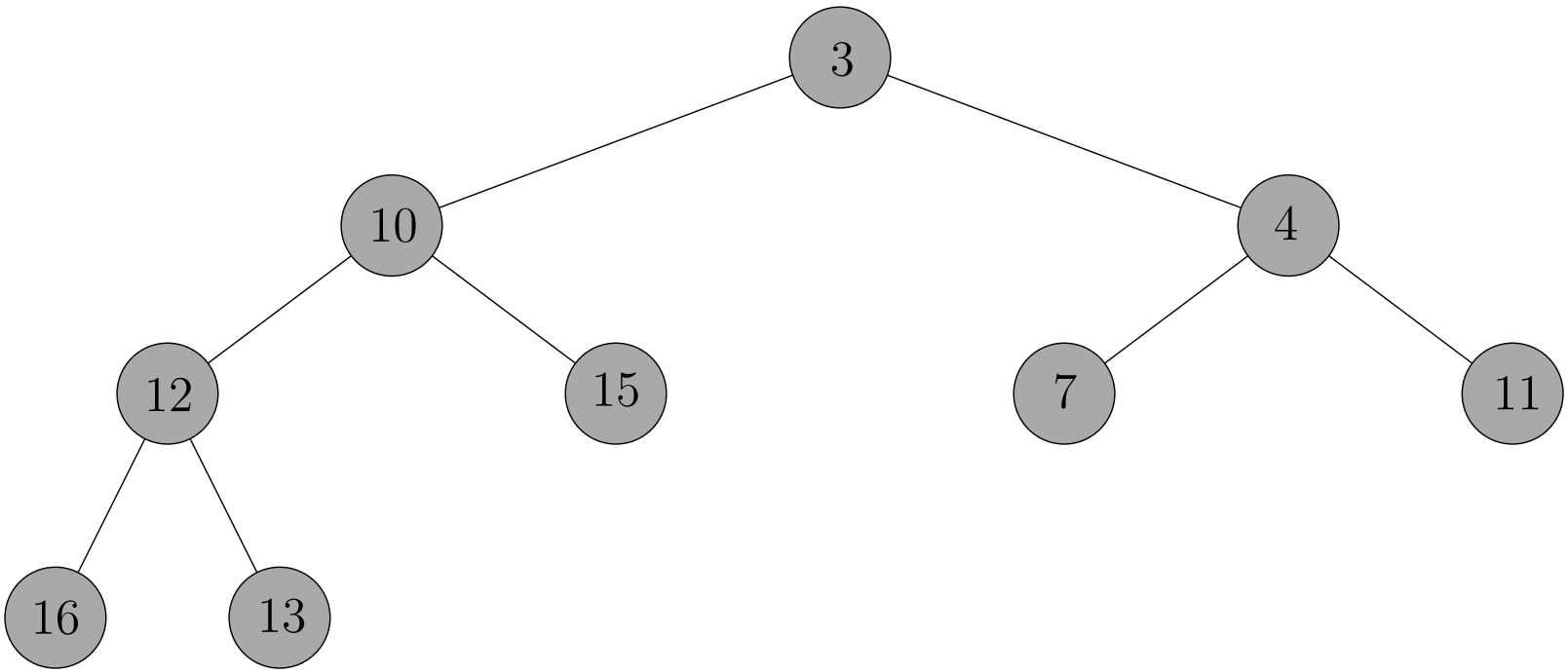
με n κόμβους



Επειδή κάθε φορά οι κόμβοι του δέντρου μοιράζονται περίπου στα δύο, ένα τέτοιο δέντρο έχει περίπου $\log n$ επίπεδα.

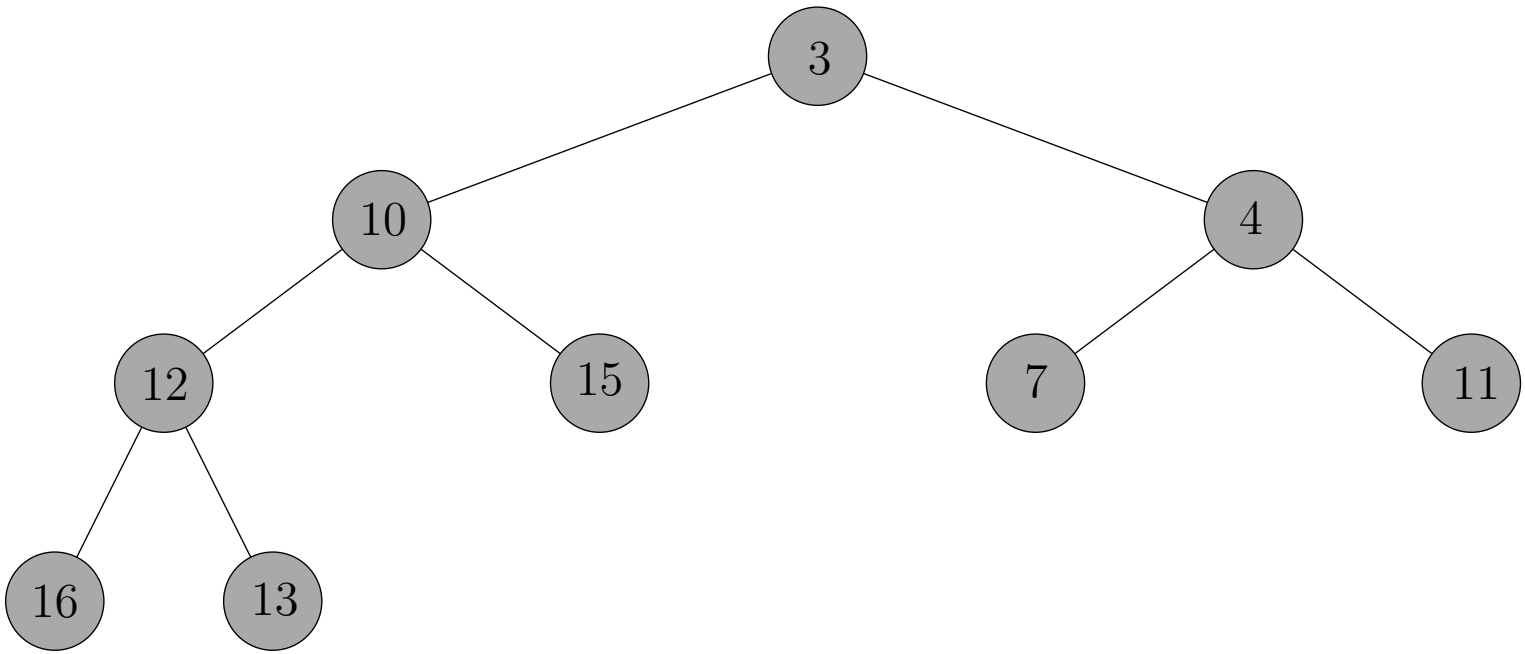
Σε αυτή την ιδιότητα βασίζονται οι δυαδικοί σωροί.

Παράδειγμα Δυαδικού Σωρού



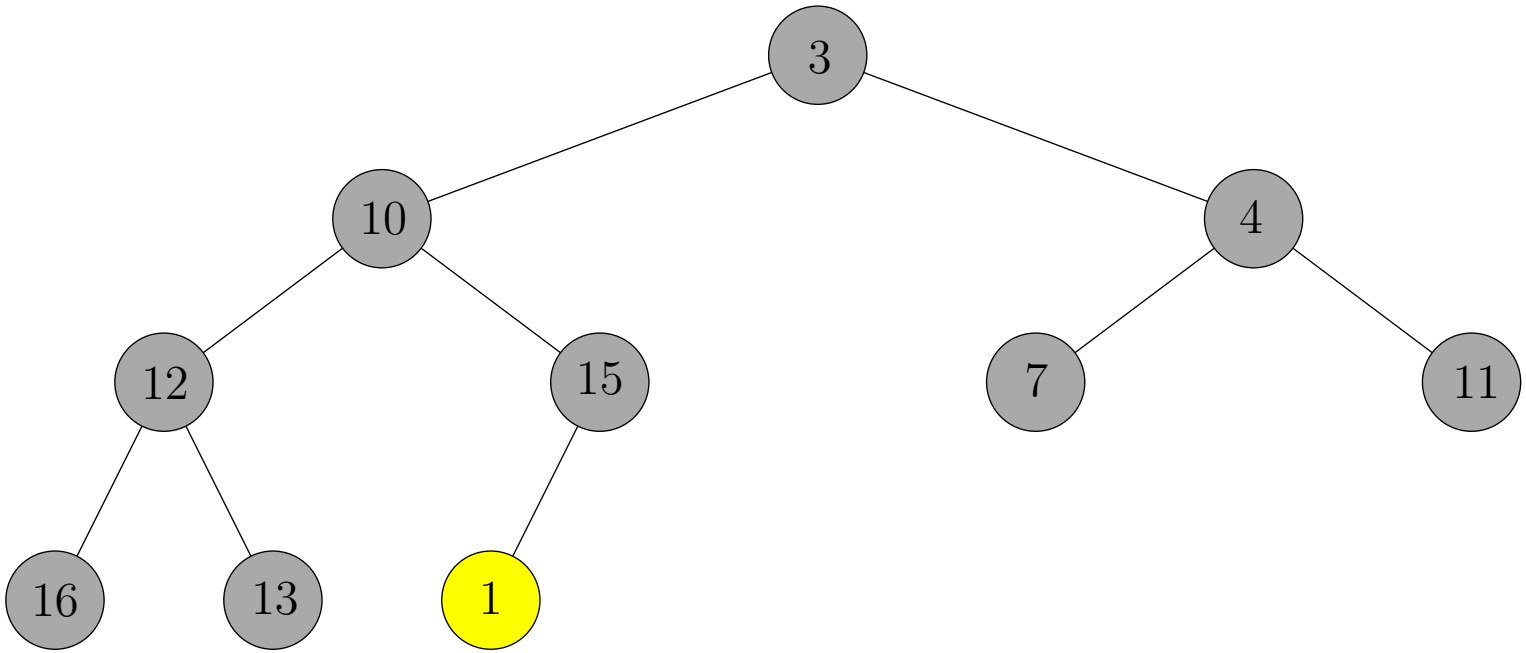
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



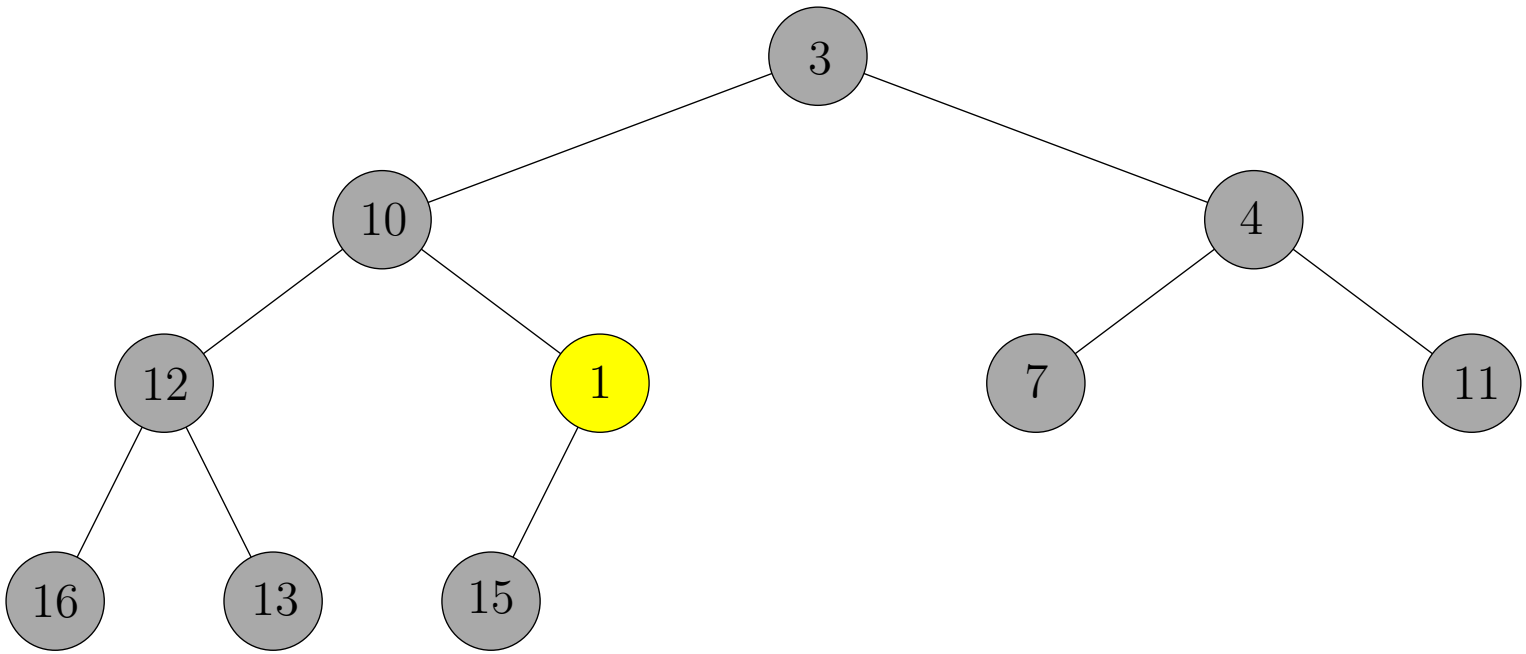
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



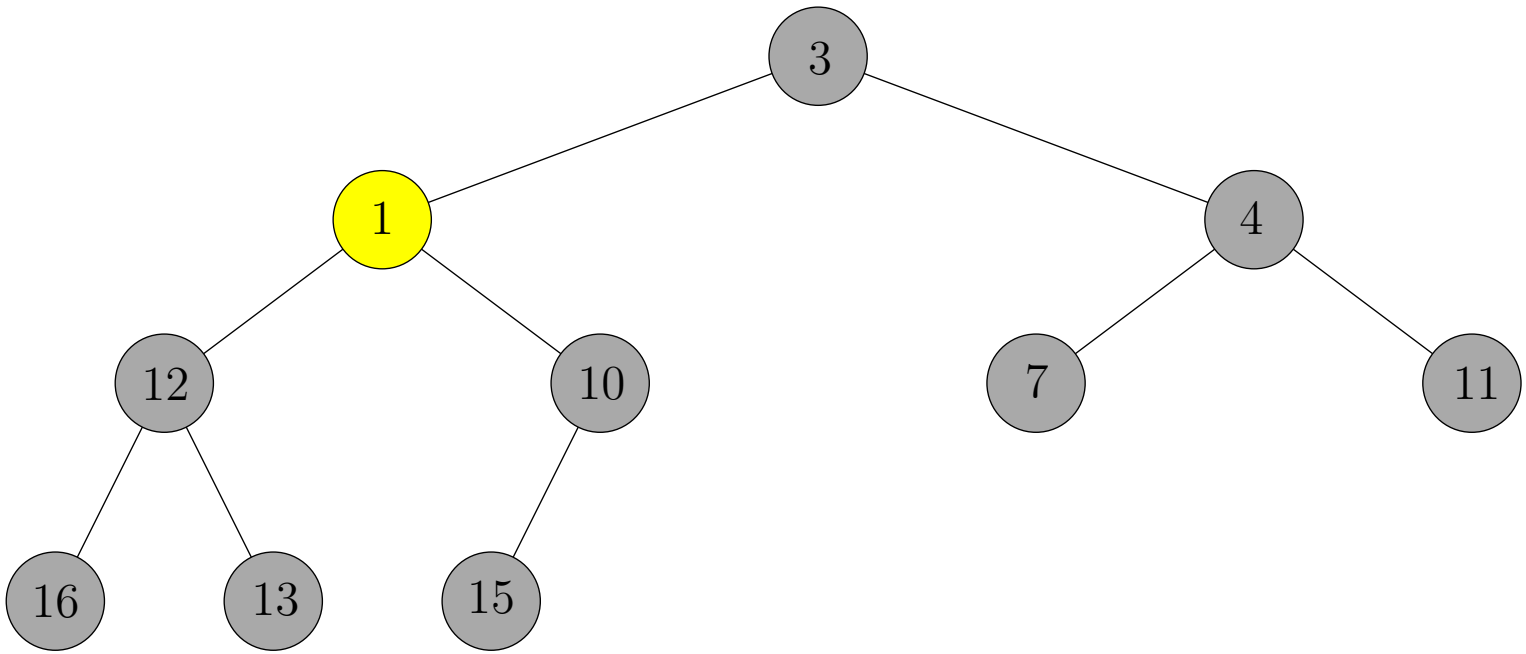
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



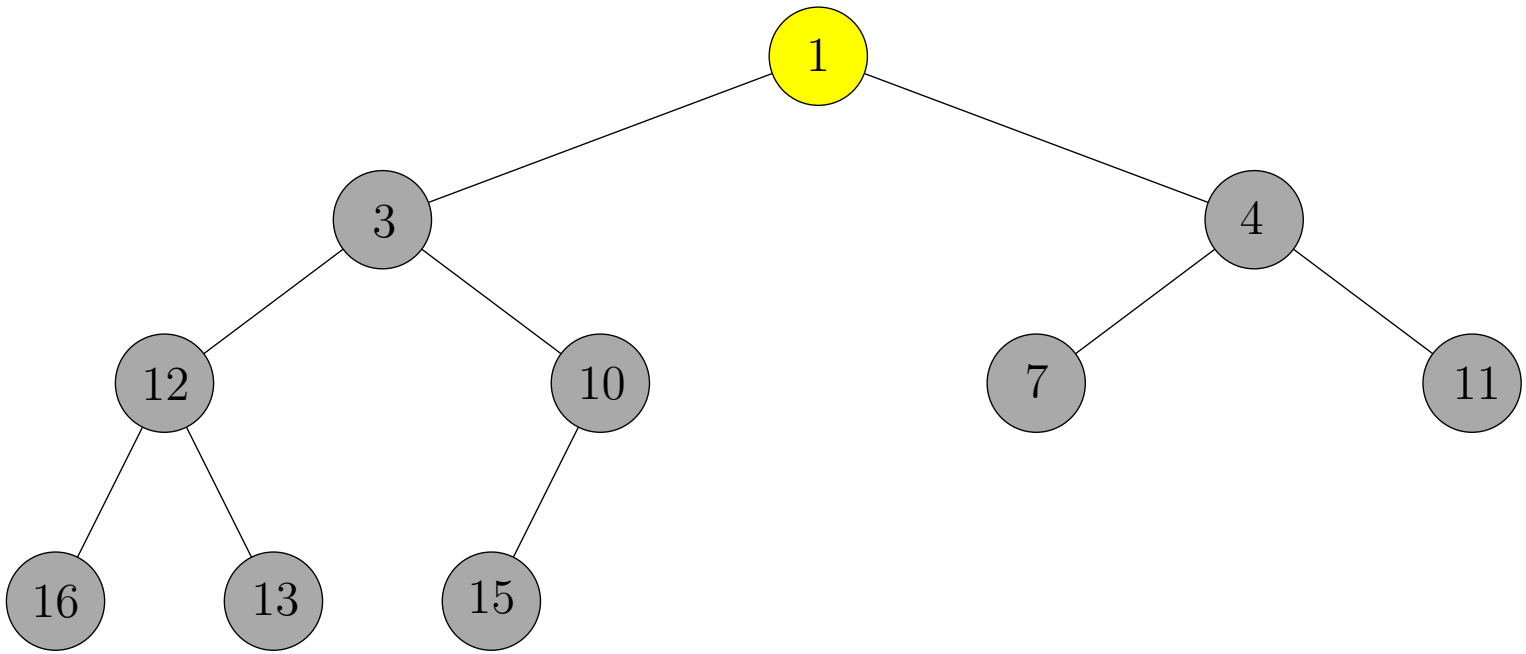
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



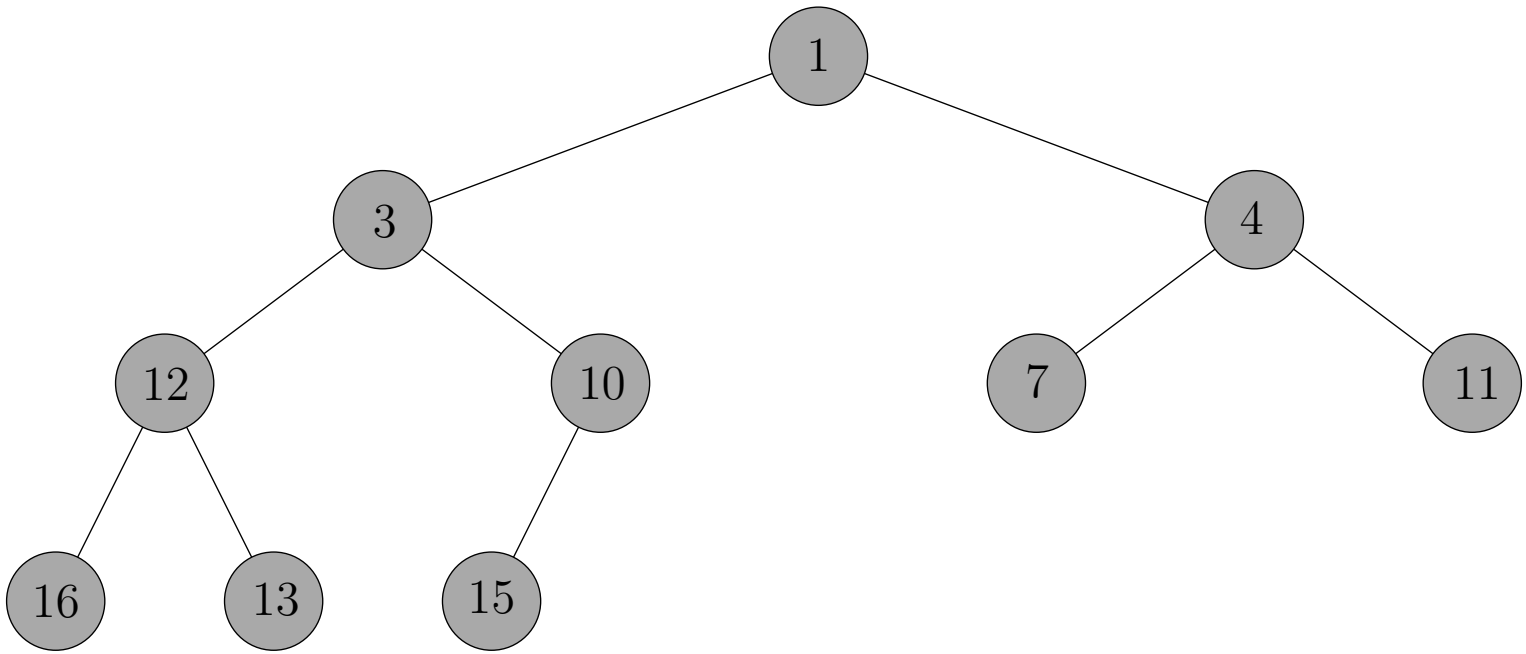
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



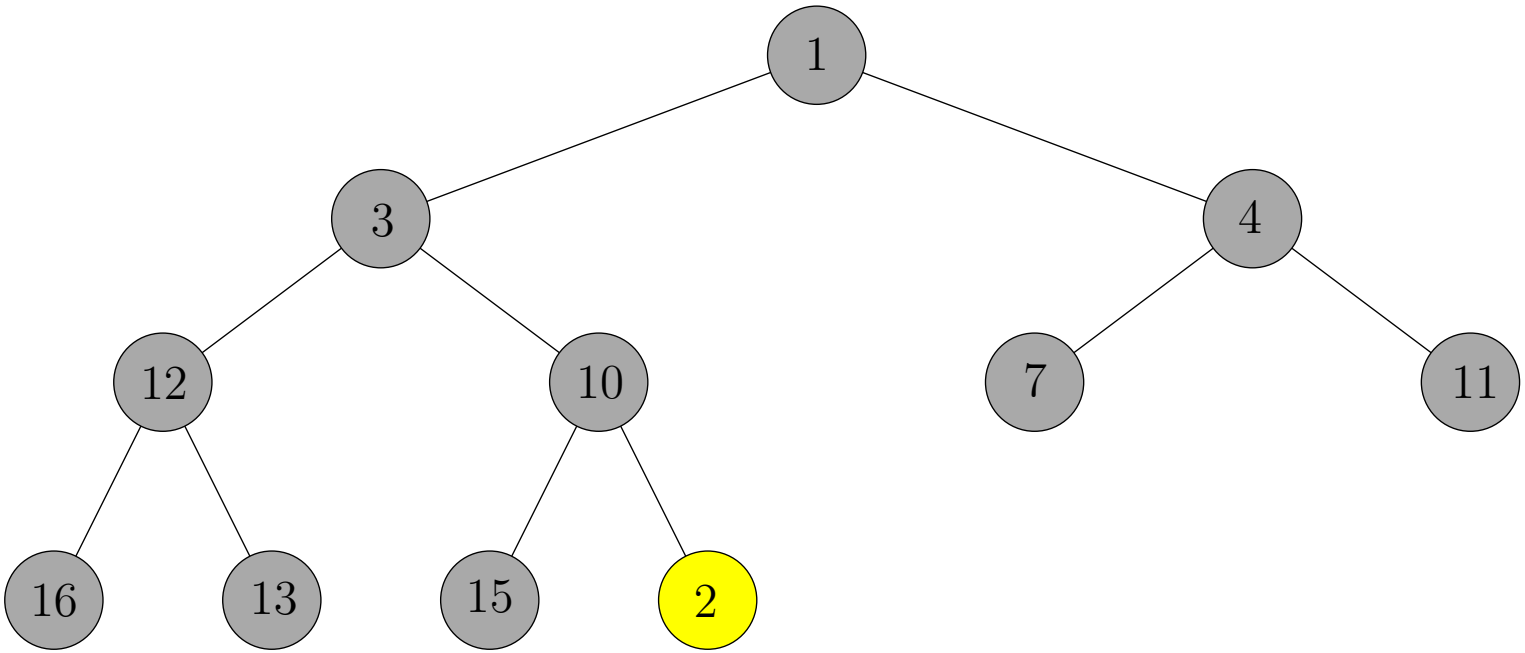
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



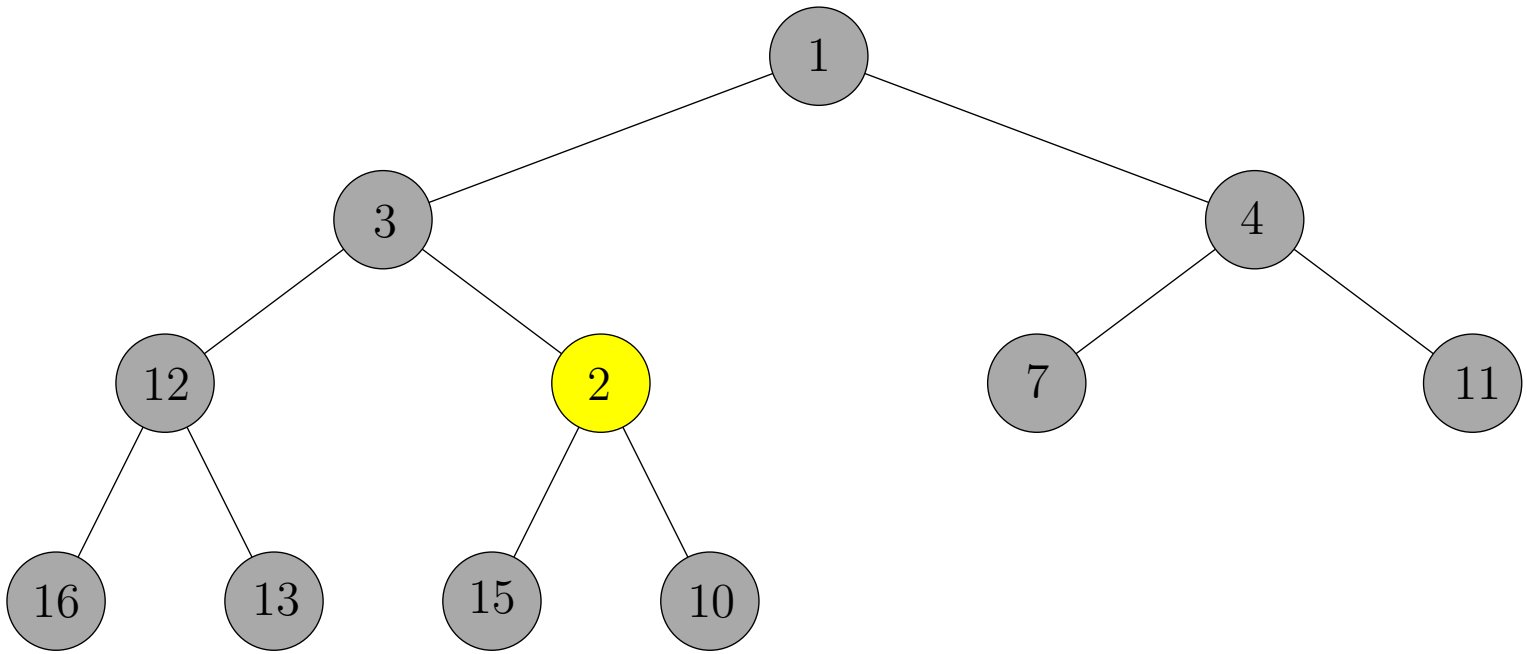
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



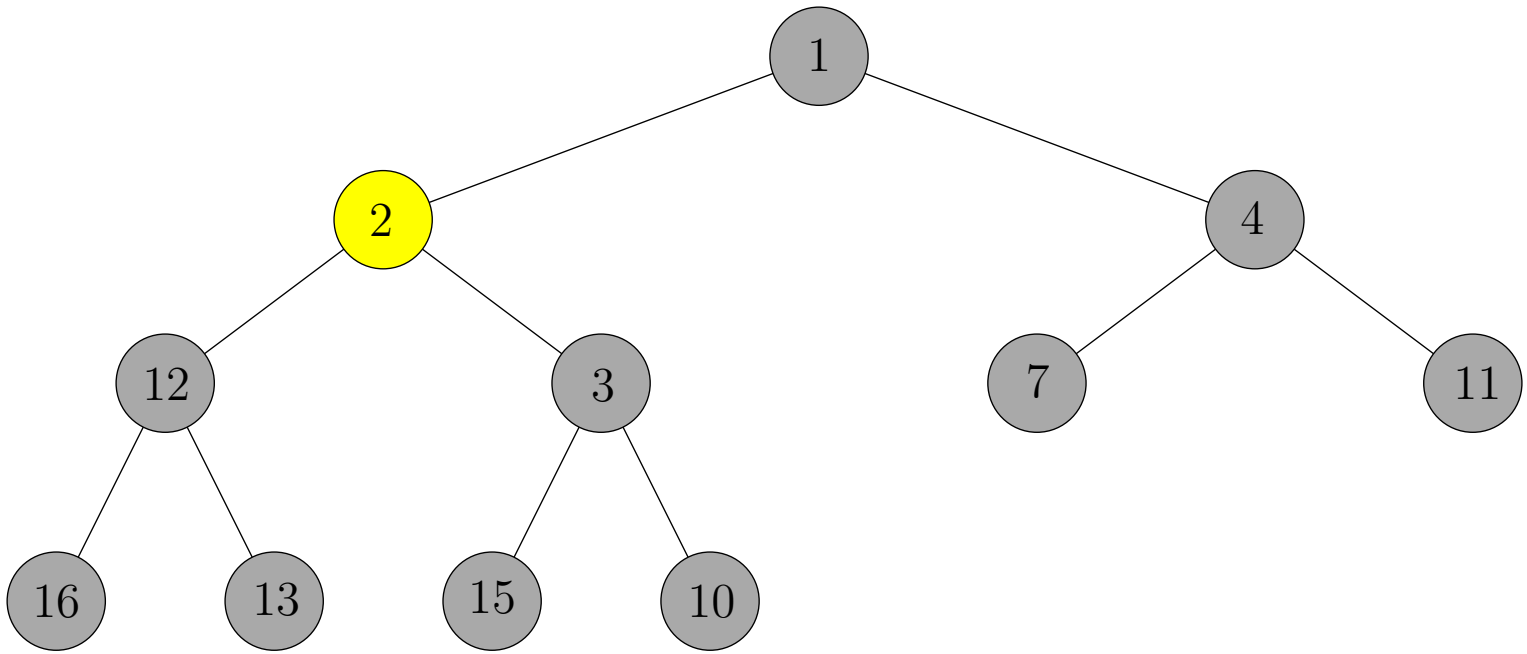
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



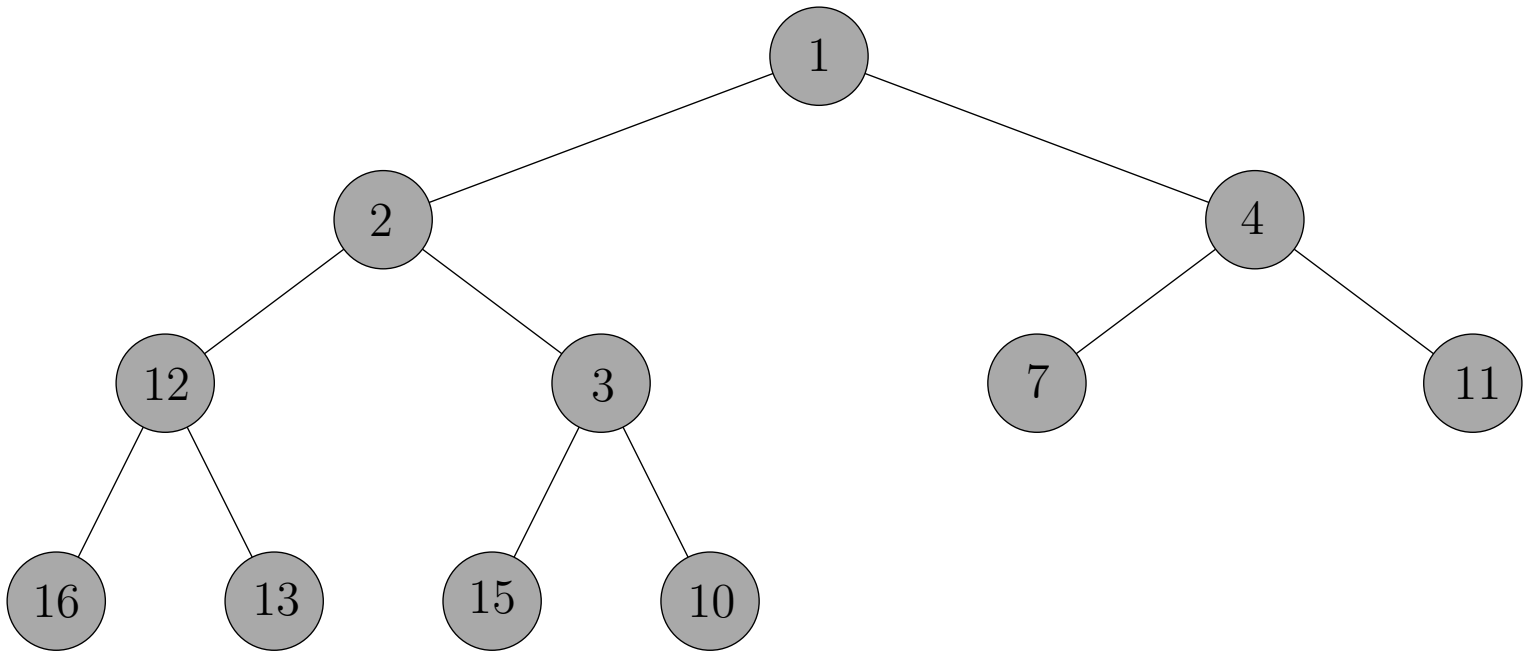
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



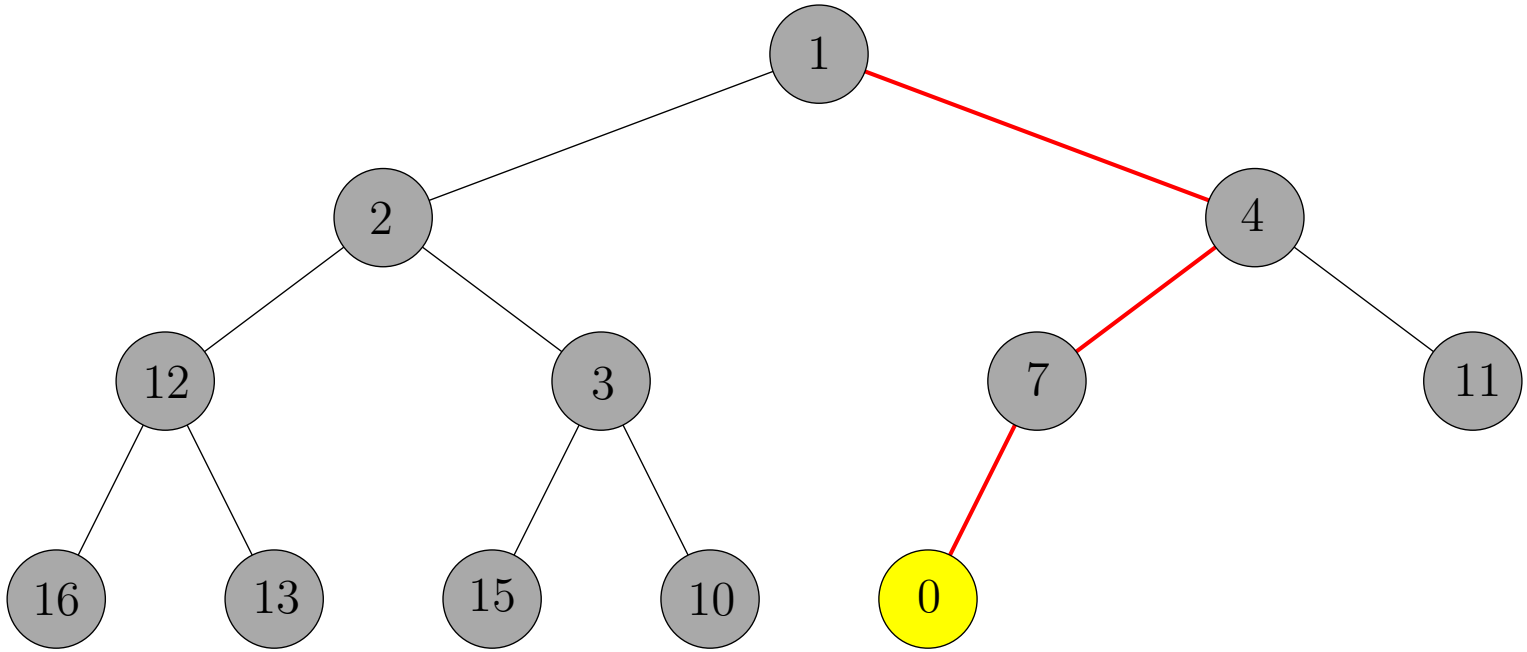
Εισαγωγή σε Δυαδικό Σωρό

κάτω-πάνω τακτοποίηση δυαδικού σωρού



Εισαγωγή σε Δυαδικό Σωρό

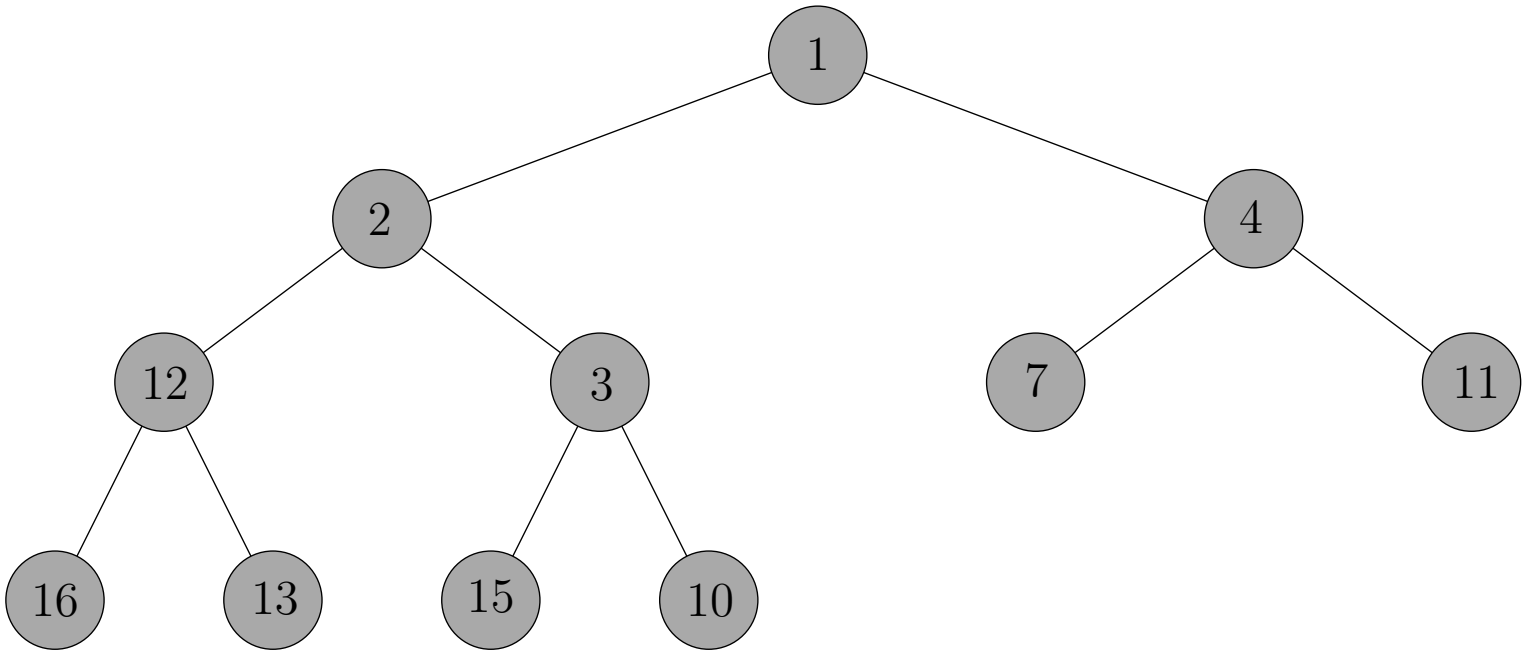
Χρόνος



Ο χρόνος εκτέλεσης είναι ανάλογος του μήκους του μέγιστου μονοπατιού από ένα φύλλο μέχρι την ρίζα, δηλαδή $O(\log n)$.

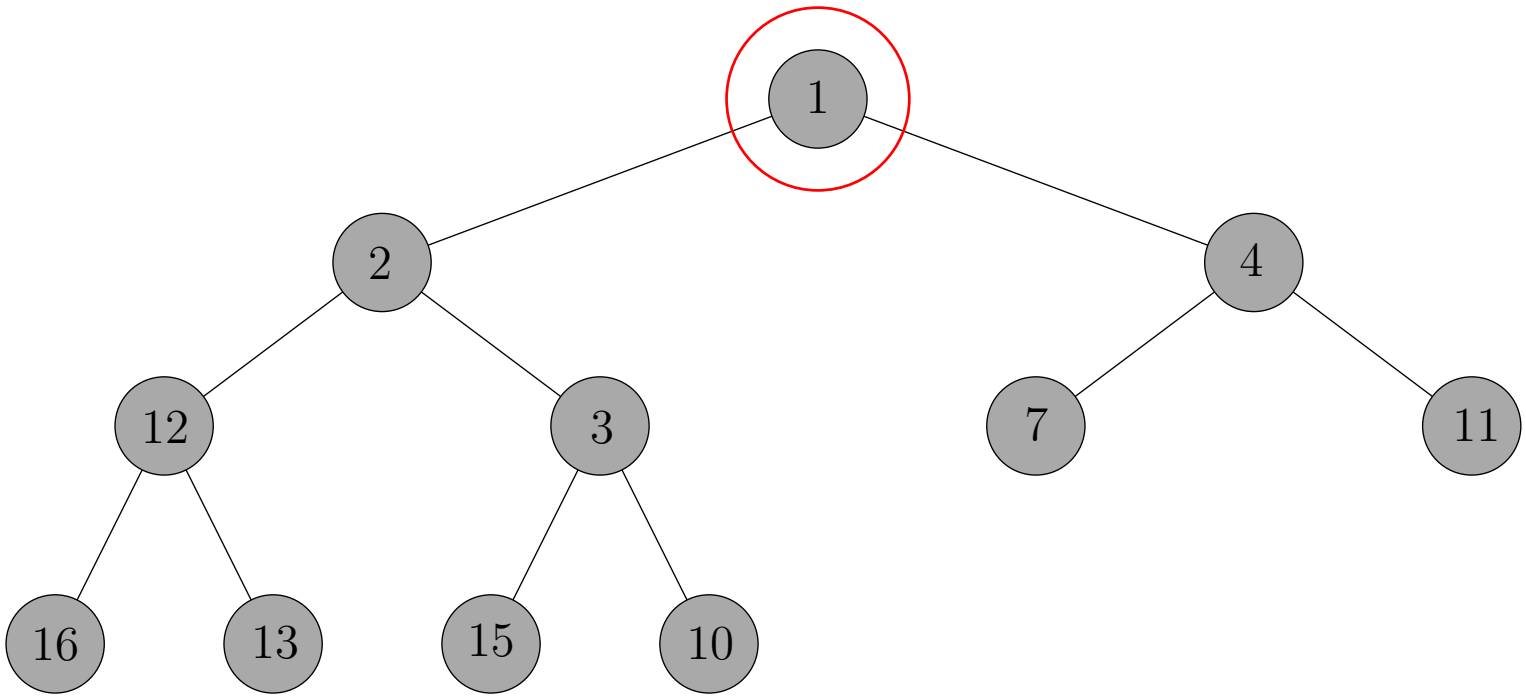
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



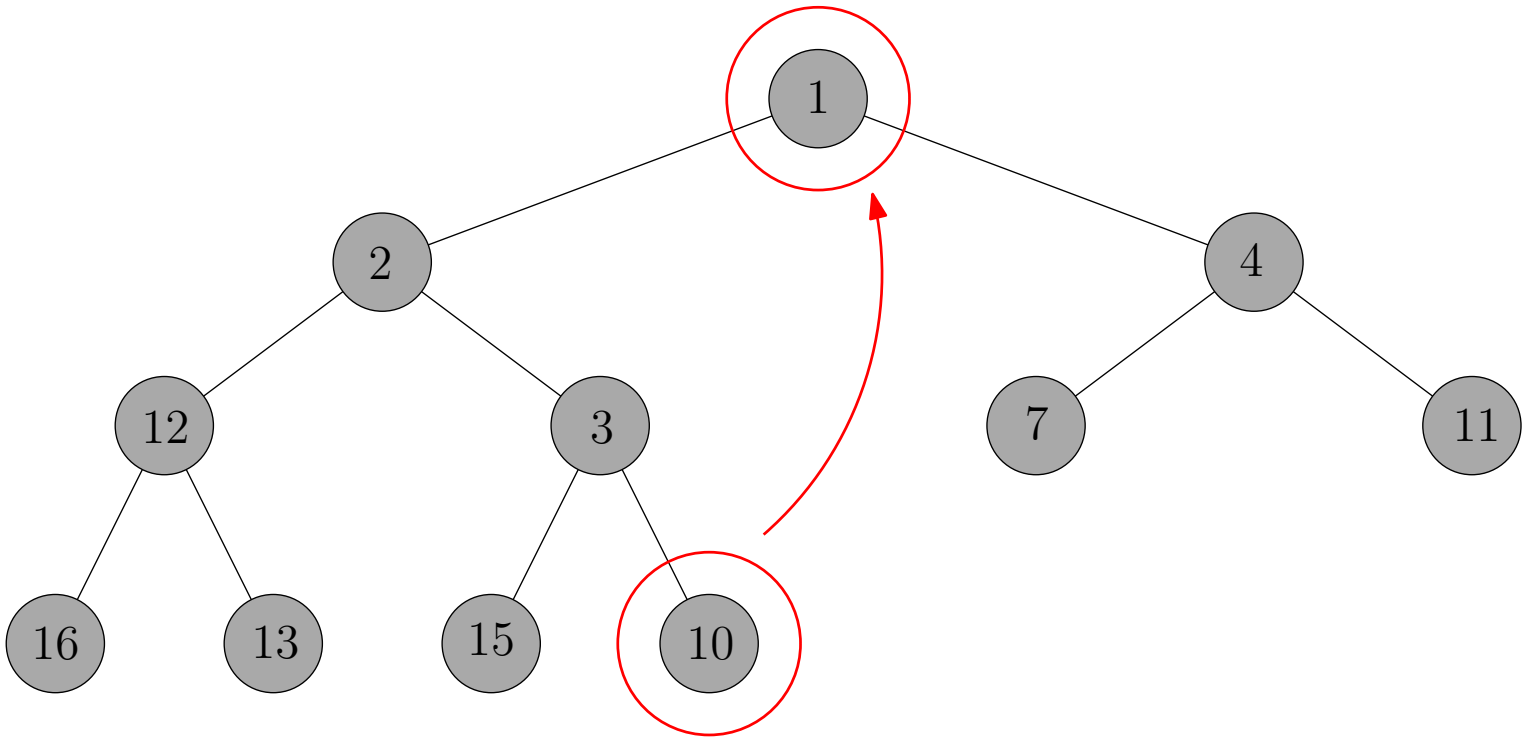
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



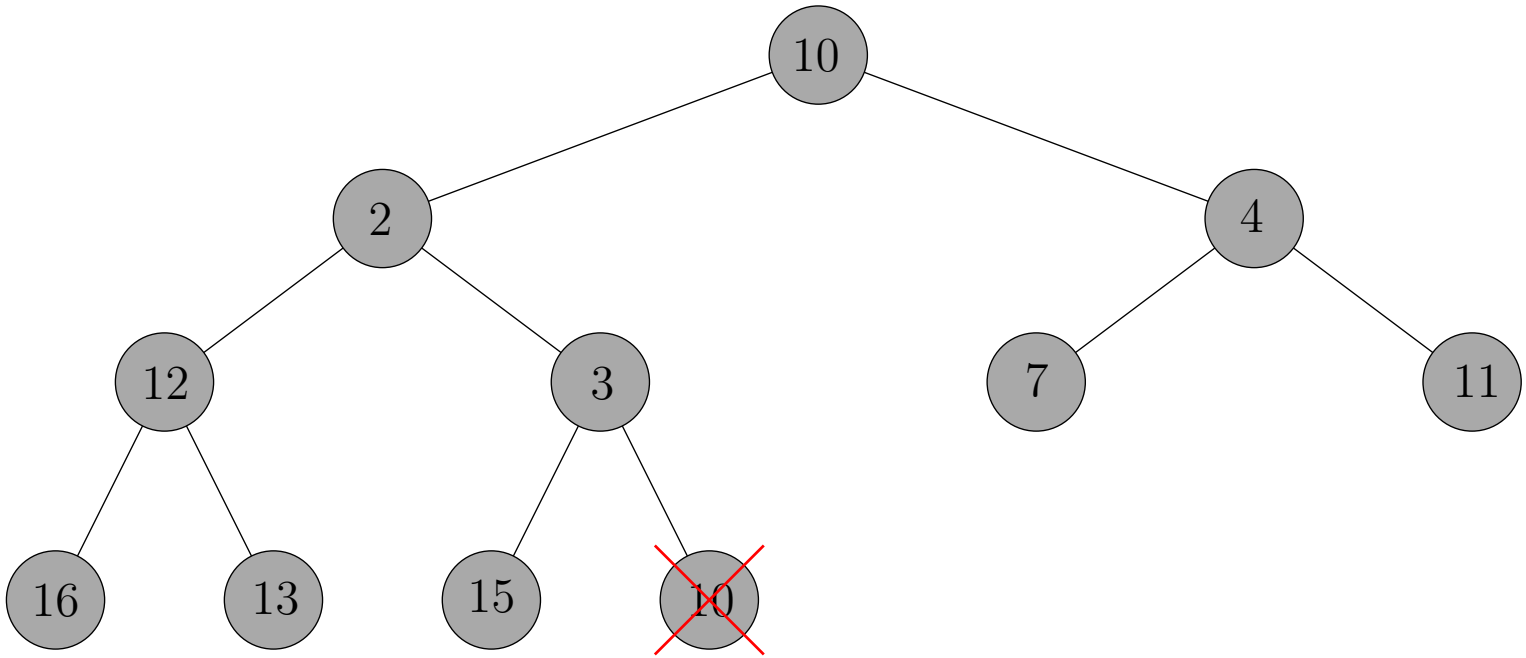
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



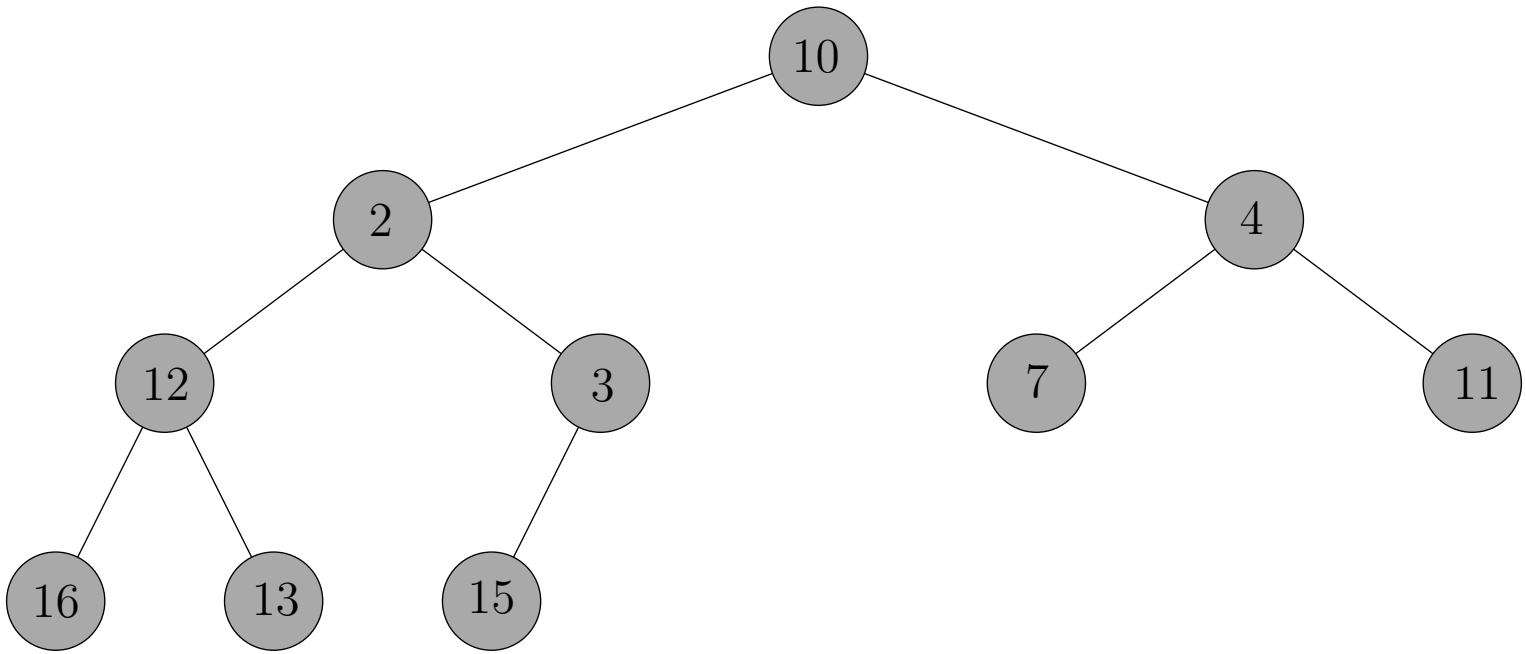
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



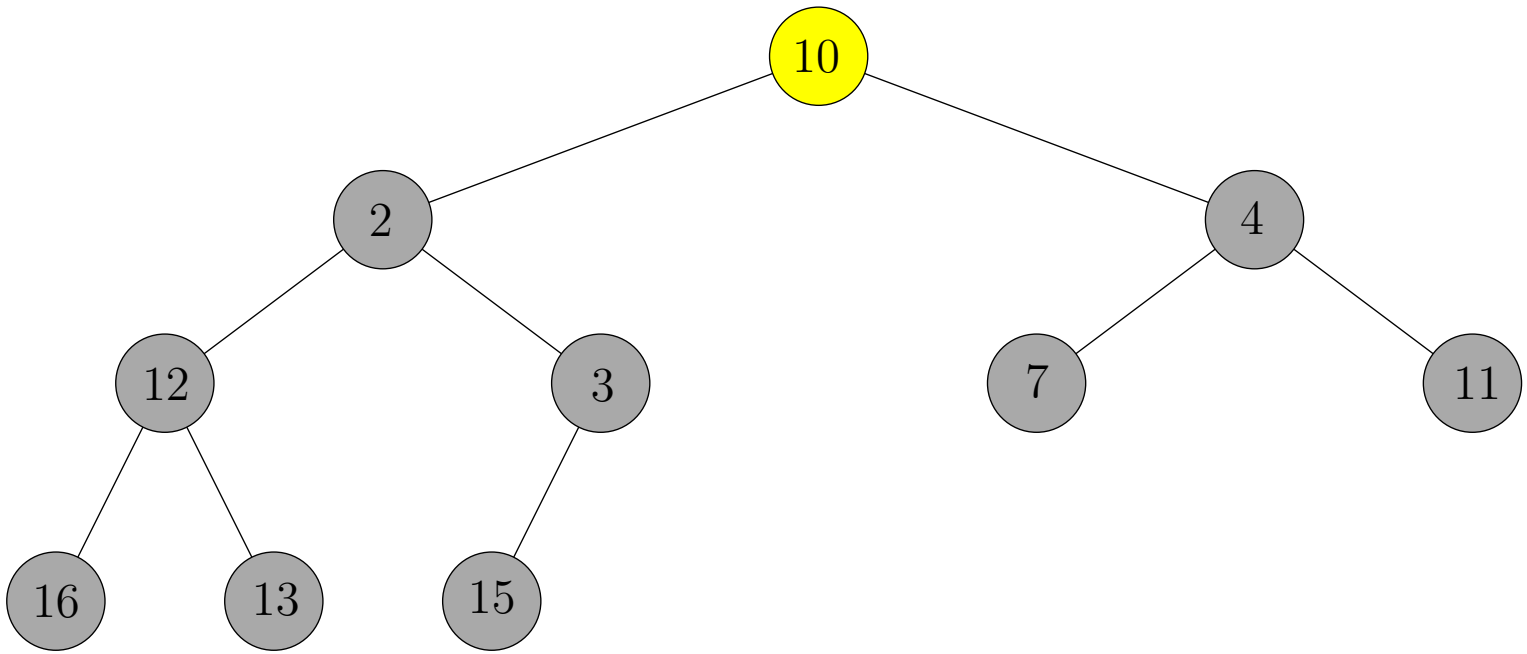
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



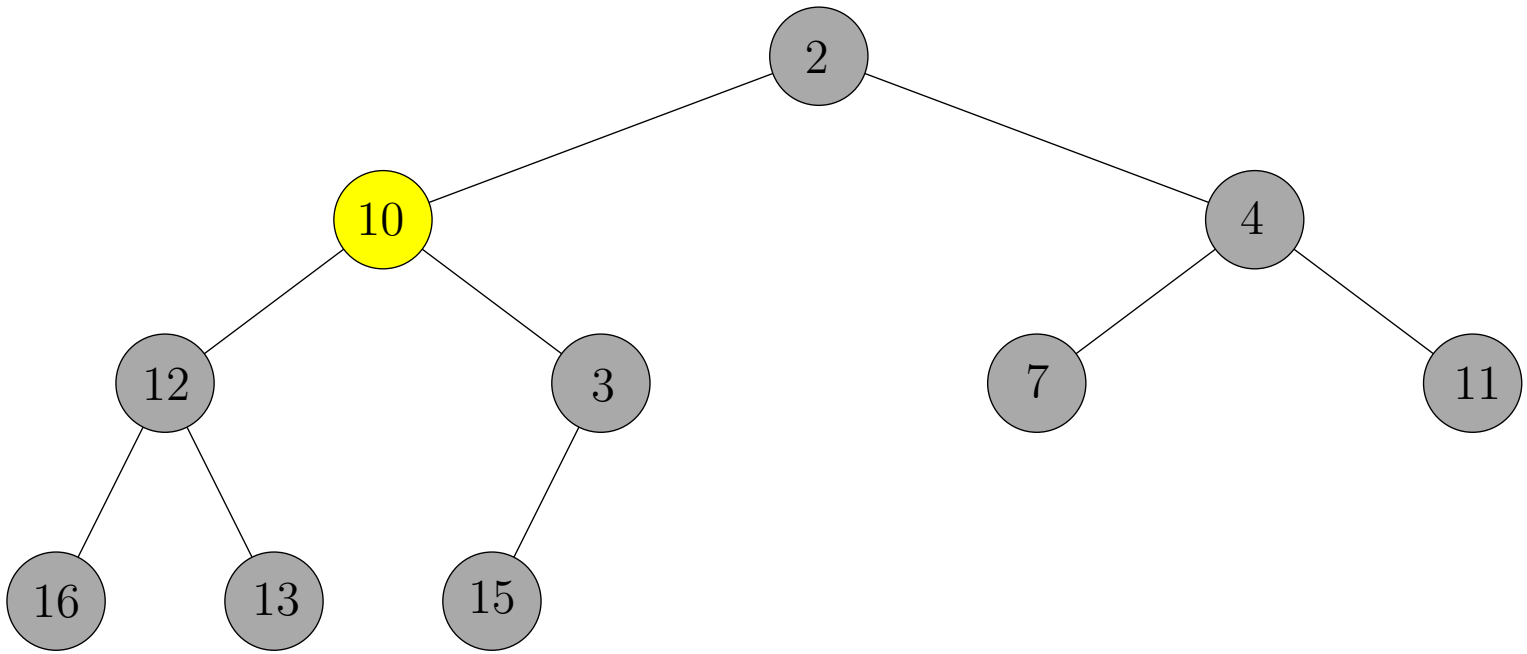
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



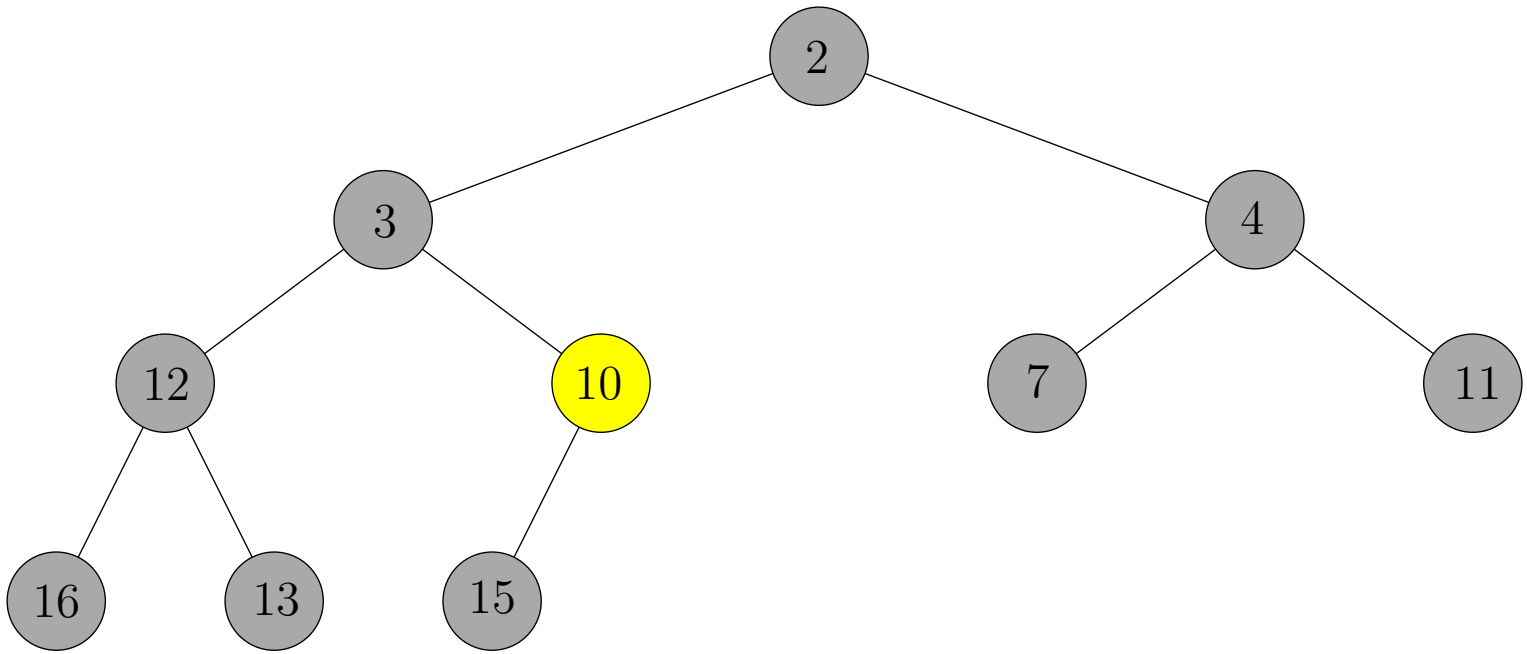
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



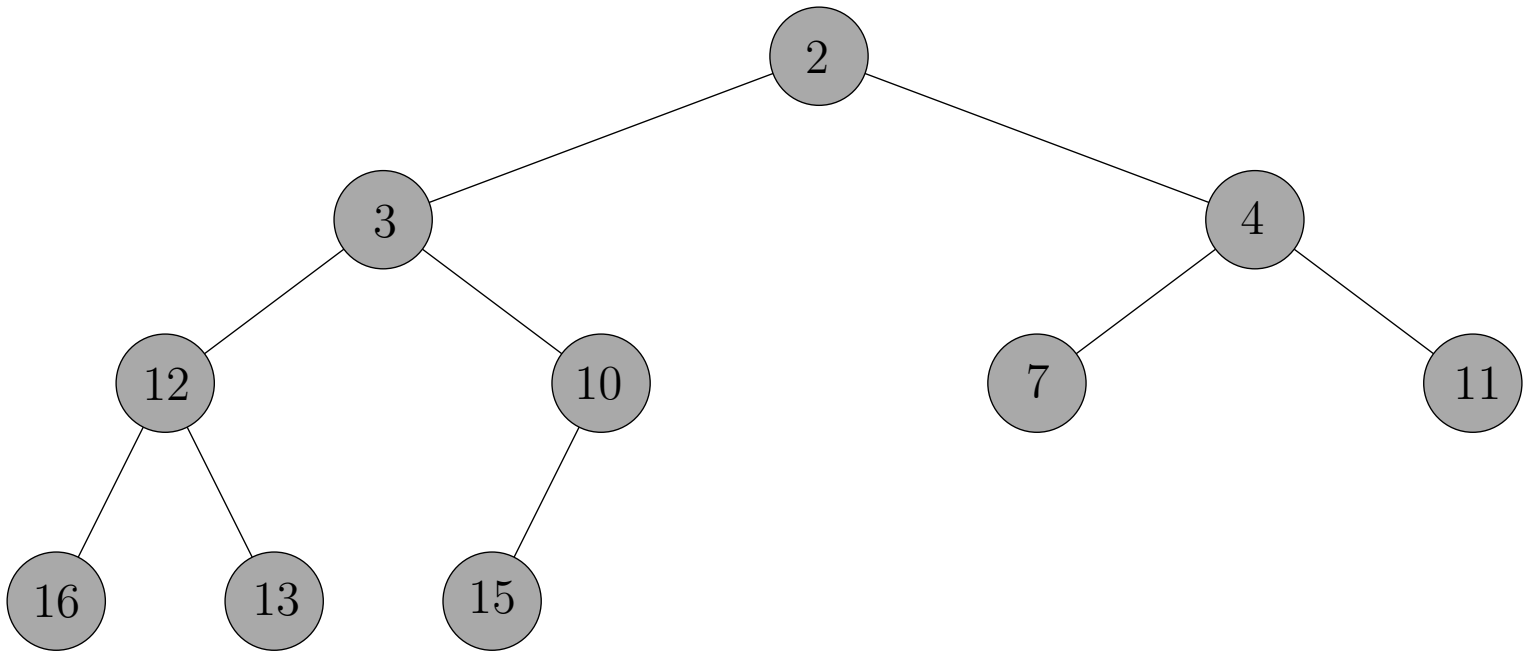
Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού

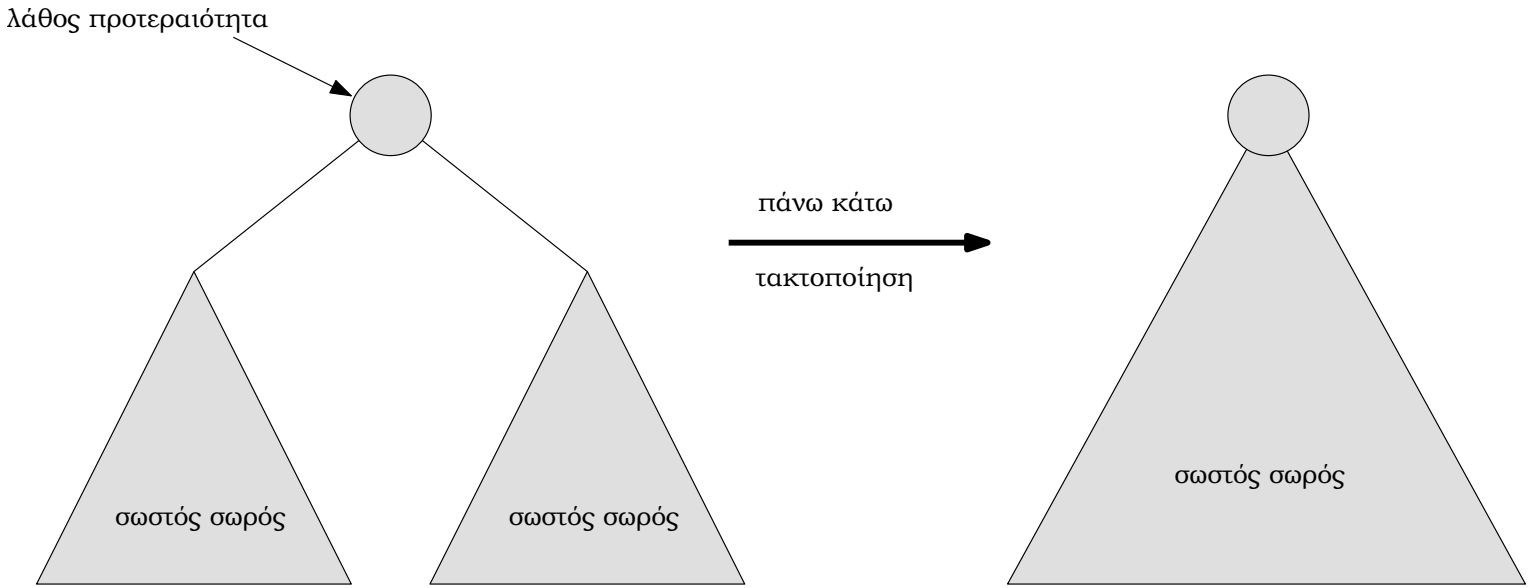


Αφαίρεση Ελαχίστου από Δυαδικό Σωρό

πάνω-κάτω τακτοποίηση δυαδικού σωρού



Πάνω-κάτω Τακτοποίηση Δυαδικού Σωρού

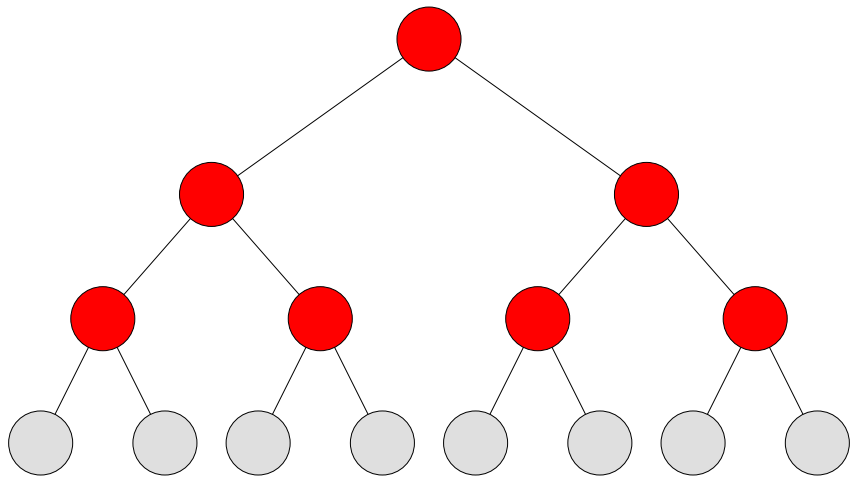


Άμα τα δύο υποδέντρα ενός κόμβου είναι σωροί, τότε με μία εκτέλεση της πάνω-κάτω τακτοποίησης σωρού από αυτόν τον κόμβο παίρνουμε ένα σωρό.

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

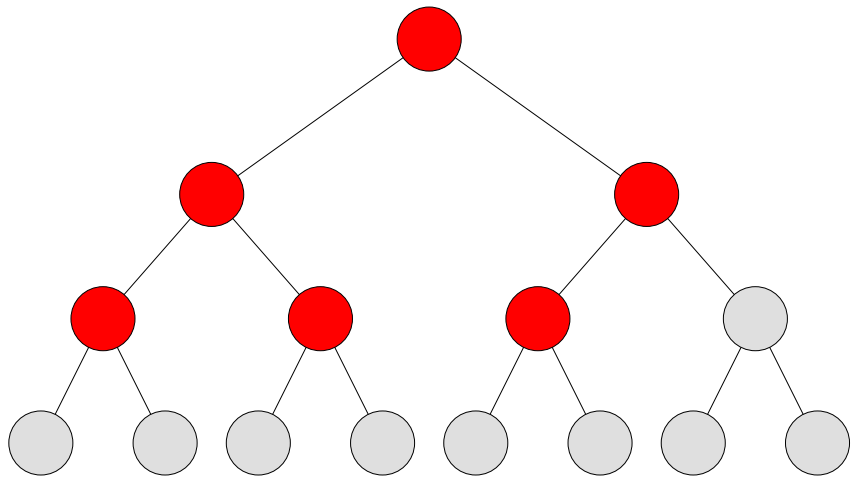


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

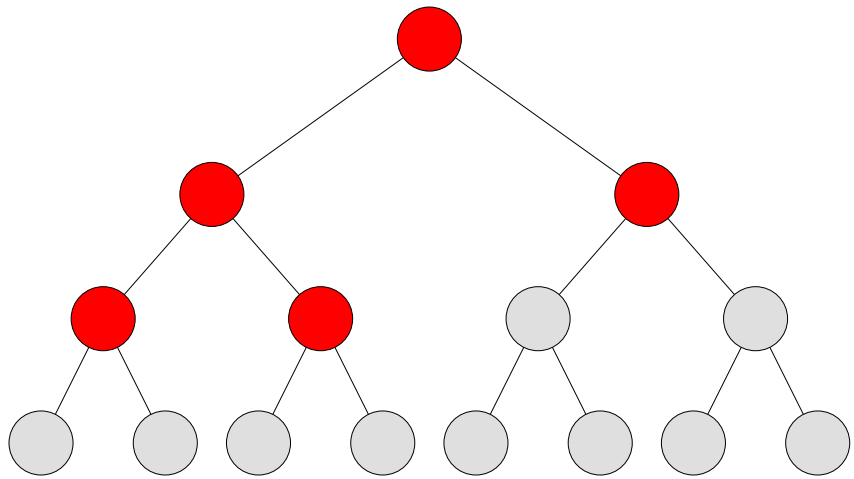


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

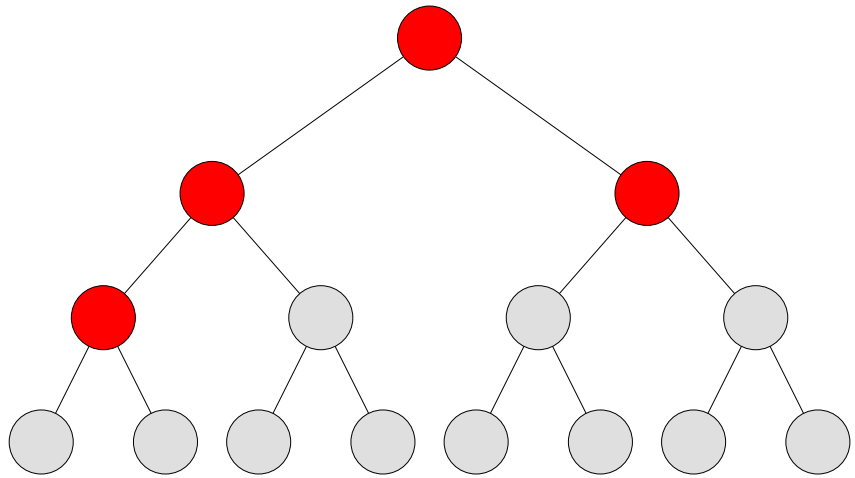


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

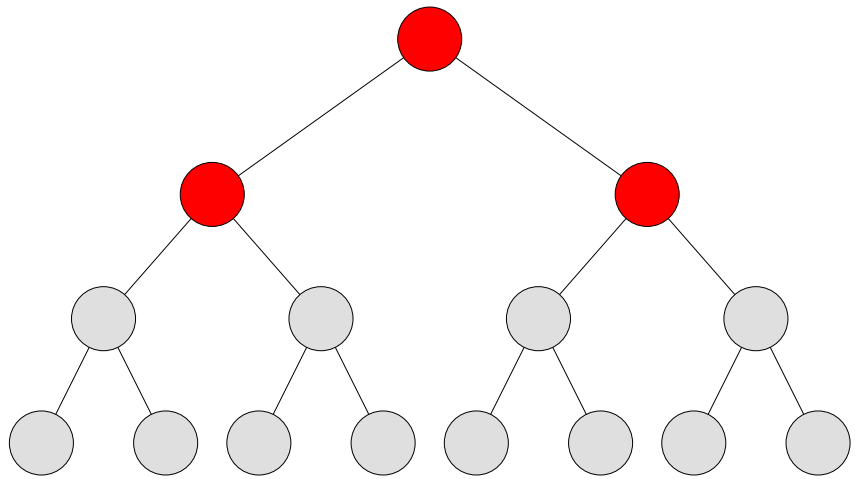


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

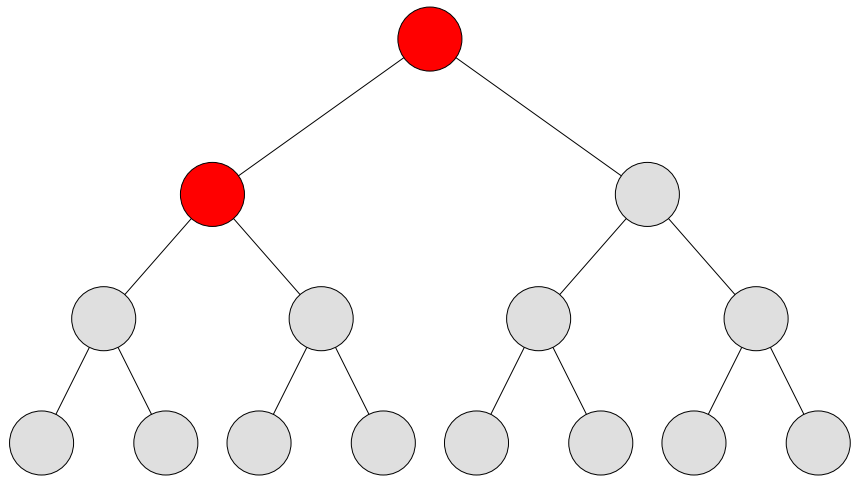


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

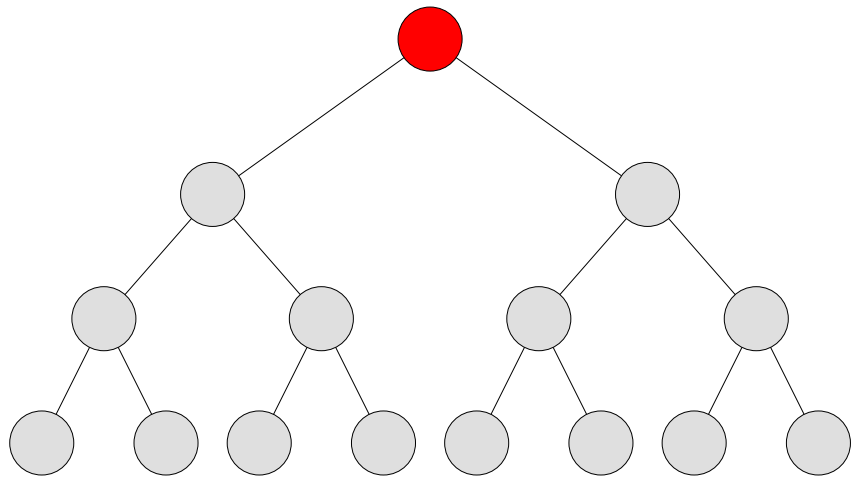


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.

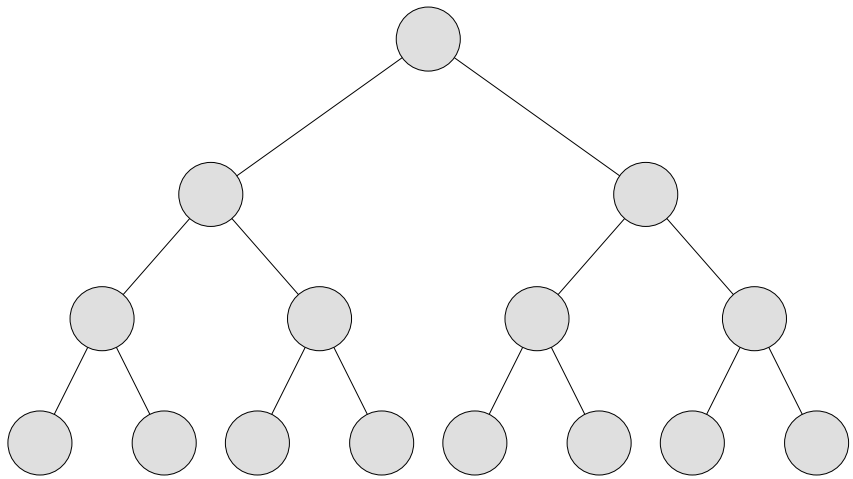


Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Έστω ένας πίνακας με n στοιχεία τον οποίο θέλουμε να μετατρέψουμε σε σωρό.



Όλα τα φύλλα του δέντρου είναι σωροί. Μπορούμε να φτιάχνουμε και το υπόλοιπο δέντρο με διαδοχικές ”πάνω-κάτω” τακτοποιήσεις. Ξεκινάμε από την μέση του πίνακα και εκτελούμε ”πάνω-κάτω” τακτοποιήσεις για ένα ένα τα στοιχεία μέχρι να φτάσουμε στην ρίζα (το πρώτο στοιχείο του πίνακα).

Κατασκευή Δυαδικού Σωρού από Πίνακα

heapify

Πόσο κοστίζει αυτή η διαδικασία;

- Ο πίνακας έχει n στοιχεία και κάνουμε ”πάνω-κάτω” τακτοποίηση το πολύ σε $k = \lfloor \log n \rfloor$ επίπεδα.
- Κάθε επίπεδο i έχει το πολύ 2^i κόμβους
- Η πάνω-κάτω τακτοποίηση ξεκινώντας από ένα κόμβο στο επίπεδο i έχει το πολύ κόστος $\mathcal{O}(k - i)$.

Προσθέτοντας:

$$\mathcal{O}\left(\sum_{0 \le i < k} 2^i (k - i)\right) = \mathcal{O}\left(2^k \sum_{0 \le i < k} \frac{k - i}{2^{k-i}}\right) = \mathcal{O}\left(2^k \sum_{j \ge 1} \frac{j}{2^j}\right) = \mathcal{O}(n)$$

Ταξινόμηση με Δυαδικό Σωρό

Έστω ένας πίνακας με n ακεραίους. Θέλουμε να τυπώσουμε τους ακεραίους από τον μικρότερο έως τον μεγαλύτερο με την σειρά.

Αλγόριθμος

- φτιάξε ένα σωρό από τον πίνακα
- όσο ο σωρός δεν είναι άδειος βγάλε το μικρότερο στοιχείο και εκτύπωσε το

Ο αλγόριθμος αυτός λέγεται Heap-Sort και βάζει n αριθμούς στη σειρά σε χρόνο $\mathcal{O}(n \log n)$.