Επαύξηση Δομών Δεδομένων

Δυναμικές Διατακτικές Στατιστικές

και

Δέντρα Διαστημάτων

Δέντρο Διατακτικής Στατιστικής

- Διατακτική Στατιστική: Η i-οστη διατακτική στατιστική ενός συνόλου n στοιχείων είναι το i-οστο μικρότερο στοιχείο του συνόλου.
- Δέντρο Διατακτικής Στατιστικής: Είναι ένα μελανέρυθρο δέντρο (reb-black tree), όπου σε κάθε κόμβο αποθηκεύεται το πλήθος των στοιχείων του υποδέντρου με ρίζα τον κόμβο αυτόν, συμπεριλαμβανομένου και του κόμβου αυτού.
- Τάξη (rank) ενός κόμβου στο δέντρο: Ορίζεται ως η θέση του κόμβου κατά τη ενδοδιατεταγμένη (inorder) διάσχιση του δέντρου

Δέντρο Διατακτικής Στατιστικής - Λειτουργίες

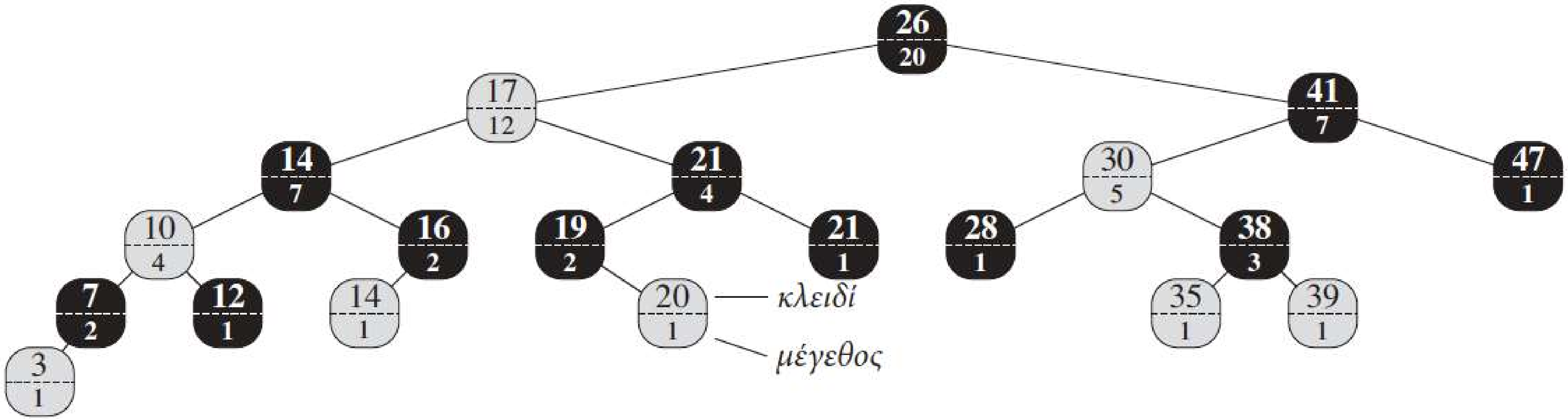
- Όλες οι λειτουργίες ενός μελανέρυθρου δέντρου
- Επιπλέον τις εξής λειτουργίες:
  - Εύρεση του κόμβου που περιέχει το i-στο μικρότερο στοιχείο του υποδέντρου με ρίζα κόμβο x
    - OS-Select(x, i)
  - Εύρεση της τάξης (rank) ενός κόμβου όπως αυτή καθορίζεται κατά την ενδοδιατεταγμένη διάσχιση του δέντρου T
    - OS_Rank(T, x)

Δέντρο Διατακτικής Στατιστικής

Επαύξηση κόμβου x:

- $x.size$: Περιέχει το πλήθος των κόμβων του υποδέντρου με ρίζα το x
- Στο πεδίο $size$ συμπεριλαμβάνεται και το ίδιο το x
- Στο πεδίο $size$ δε συμπεριλαμβάνονται οι εξωτερικοί κόμβοι (φύλλα)
- Για ένα κόμβο x ισχύει:
  – $x.size = 0$, αν x εξωτερικός κόμβος
  – $x.size = x.left.size + x.right.size + 1$, αν x εσωτερικός κόμβος

Δέντρο Διατακτικής Στατιστικής



Επαυξημένο Red-Black δέντρο

Ανάκτηση i-οστου μικρότερου κλειδιού

OS-Select(x, i)

$1. r = x.left.size + 1$

$\mathbf{2. if}\ i == r$

$3. \quad return\ x$

$\mathbf{4. else\ if}\ i < r$

$\mathbf{5.} \quad \mathbf{return}\ OS - Select(x.left, i)$

6. else

$\mathbf{7.} \quad \mathbf{return}\ OS - Select(x.right, i - r)$

Ανάκτηση i-οστου μικρότερου κλειδιού

Ορθότητα αλγορίθμου OS-Select(x, i):

- $r =$ η τάξη του κόμβου x στο υποδέντρο με ρίζα τον x
- Αν $i = r \Rightarrow x$ είναι το στοιχείο που ψάχνουμε
- Αν $i < r \Rightarrow$
  - το i-στο μικρότερο στοιχείο βρίσκεται στο αριστερό υποδέντρο του $x \Rightarrow$
  - αναζητούμε το i-στο μικρότερο στοιχείο στο υποδέντρο αυτό
- Αν $i > r \Rightarrow$
  - το i-στο μικρότερο στοιχείο βρίσκεται στο δεξί υποδέντρο του x.
  - Θα πρέπει όμως να αναζητήσουμε το $(i - r)$-στο στοιχείο στο δεξί υποδέντρου το x, αφού θα πρέπει να αφαιρέσουμε τα μικρότερα στοιχεία που βρίσκονται στο υποδέντρο με ρίζα το x.

Ανάκτηση i-οστου μικρότερου κλειδιού

Πολυπλοκότητα αλγορίθμου OS-Select(x, i):

- Μετά από κάθε αναδρομική κλήση του OS-Select κατεβαίνουμε ένα επίπεδο (ως προς το ύψος του δέντρου)
- Τα μελανέρυθρα δέντρα είναι ισορροπημένα, έχουν δηλαδή ύψος $O(\lg n)$
- Άρα θα έχουμε το πολύ $O(\lg n)$ κλήσεις του OS-Select
- Άρα η πολυπλοκότητα είναι $O(\lg n)$

Εύρεση Τάξης (Rank) Στοιχείου

OS-Rank(T, x)

$r = x.left.size + 1$

$y = x$

$\textbf{while}\ y \neq T.root$

$\quad \textbf{if}\ y == y.p.right$

$\qquad r = r + y.p.left.size + 1$

$\quad y = y.p$

$\textbf{return}\ r$

Εύρεση Τάξης (Rank) Στοιχείου - Ορθότητα

Αναλλοίωτη συνθήκη OS-Rank(T, x): Στην έναρξη κάθε επανάληψης του βρόχου (while loop, γραμμή 3), το r είναι η τάξη του x, στο υποδέντρο με ρίζα το y.

Αρχικός έλεγχος:

- Στην αρχή της πρώτης επανάληψης $r \leftarrow$ τάξη του x, στο υποδέντρο με ρίζα το x.
- Με την ανάθεση $y \leftarrow x$, εξασφαλίζεται η συνθήκη.

Εύρεση Τάξης (Rank) Στοιχείου - Ορθότητα

Αναλλοίωτη συνθήκη OS-Rank(T, x): Στην έναρξη κάθε επανάληψης του βρόχου (while loop γραμμές 3 - 6), το r είναι η τάξη του x, στο υποδέντρο με ρίζα το y.

Έλεγχος Διατήρησης:

- Στο τέλος κάθε επανάληψης του βρόχου, ισχύει $y \leftarrow y.p$
- Άρα πρέπει να δείξουμε ότι:
  - αν $r =$ τάξη του x στο υποδέντρο με ρίζα y στην αρχή του βρόχου $\Rightarrow$ το r θα είναι τάξη του x στο τέλος του βρόχου στο υποδέντρο με ρίζα $y.p$
- Δύο περιπτώσεις:
  - Αν $y == y.p.left$ $\Rightarrow$ το r μένει αναλλοίωτο $\Rightarrow$ η συνθήκη ισχύει
  - Αν $y == y.p.right$ $\Rightarrow$ οι κόμβοι που βρίσκονται στο υποδέντρο με ρίζα τον $y.p.left$ προηγούνται (έχουν μικρότερη τάξη) όλων των κόμβων στο υποδέντρο με ρίζα το y $\Rightarrow$
  - προσθέτουμε στο r το πλήθος των κόμβων (size) του υποδέντρου με ρίζα το $y.p.left$ καθώς και 1 για το ίδιο το $y.p$

Εύρεση Τάξης (Rank) Στοιχείου - Ορθότητα

Αναλλοίωτη συνθήκη OS-Rank(T, x): Στην έναρξη κάθε επανάληψης του βρόχου (while loop γραμμές 3 - 6), το r είναι η τάξη του x, στο υποδέντρο με ρίζα το y.

Επιβεβαίωση αποτελέσματος (τερματισμός):

- Σε κάθε επανάληψη το y μετακινείται ένα επίπεδο πάνω, μέχρι τελικά τη ρίζα του δέντρου, δηλαδή όταν $y = T.root$.
- Όταν τερματίσει ο βρόχος (while loop γραμμές 3 - 6) το υποδέντρο με ρίζα το y είναι όλο το δέντρο $T \Rightarrow$ $r =$ τάξη του x στο δέντρο T.

Εύρεση Τάξης (Rank) Στοιχείου

Πολυπλοκότητα OS-Rank(T, x):

- Σε κάθε επανάληψης του βρόχου το r ανεβαίνει ένα επίπεδο, $y = y.p$
- Το ύψος του δέντρου είναι $O(\lg n)$
- Άρα η πολυπλοκότητα: $O(\lg n)$

Διατήρηση-Ενημέρωση Κόμβων – Εισαγωγή

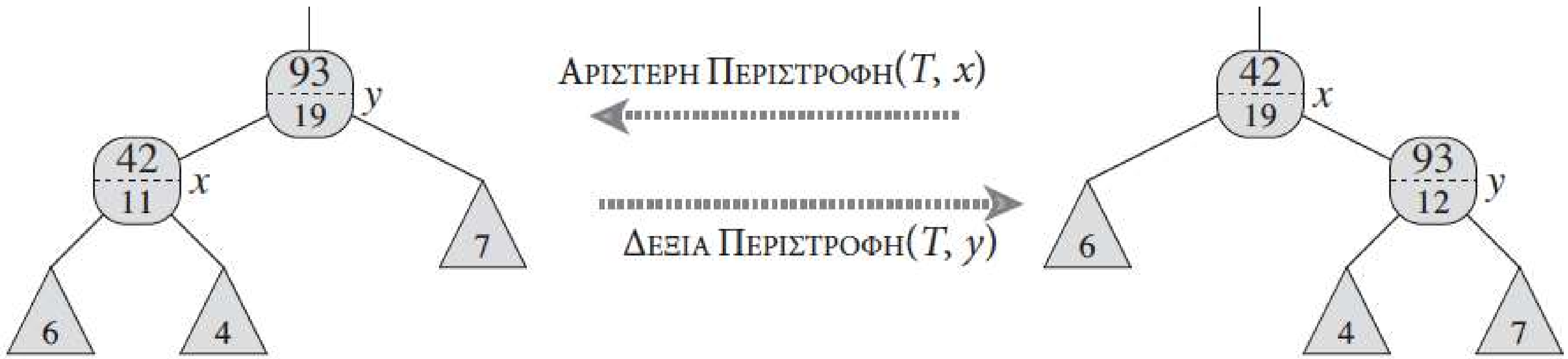
Βήμα 1:

- Ενημερώνουμε το πεδίο size κάθε κόμβου που βρίσκεται στο μονοπάτι από τη ρίζα μέχρι το κόμβο στον οποίο θα προστεθεί ως παιδί ο νέος κόμβος ως εξής:
- $size = size + 1$

Διατήρηση-Ενημέρωση Κόμβων – Εισαγωγή

Βήμα 2:

- Ανεβαίνουμε προς τη ρίζα για να διορθώσουμε πιθανές παραβιάσεις των ιδιοτήτων του μελανέρυθρου δέντρου
  - αλλαγή χρώματος: δεν επηρεάζει το πεδίο $size$ των κόμβων
  - Περιστροφές (#περιστοφών ≤ 2): επηρεάζει το πεδίο $size$ των κόμβων
- Διόρθωση πεδίων size κατά την αριστερή περιστροφή:
  - $y.size = x.size$
  - $x.size = x.left.size + x.right.size + 1$
  - Συμμετρικές είναι οι σχέσεις για δεξιά περιστροφή



Διατήρηση-Ενημέρωση Κόμβων – Εισαγωγή

Πολυπλοκότητα

- $\mathrm{O}(1)$ για κάθε περιστροφή $\Rightarrow$ $\mathrm{O}(1)$ για ενημέρωση του πεδίου $size$
- Άρα η πολυπλοκότητα εισαγωγής παραμένει $O(\lg n)$

Διατήρηση-Ενημέρωση Κόμβων – Διαγραφή

Βήμα 1:

- Διαγραφή του κόμβου από το δέντρο
- Διάσχιση του μονοπατιού από τον κόμβο προς διαγραφή και ενημέρωση του πεδίου $size$
- $size = size - 1$
- Ύψος δέντρου $O(\lg n) \Rightarrow$ πολυπλοκότητα βήματος $O(\lg n)$

Διατήρηση-Ενημέρωση Κόμβων – Διαγραφή

Βήμα 2:

- Ανεβαίνουμε προς τη ρίζα για να διορθώσουμε πιθανές παραβιάσεις των ιδιοτήτων του μελανέρυθρου δέντρου
- αλλαγή χρώματος: δεν επηρεάζει το πεδίο $size$ των κόμβων
- Περιστροφές (#περιστοφών ≤ 3): επηρεάζει το πεδίο $size$ των κόμβων
- Όμοια με την περίπτωση των περιστροφών κατά την εισαγωγή η πολυπλοκότητα μιας περιστροφής $\Rightarrow O(1)$
- Συνολική πολυπλοκότητα διαγραφής: $O(\lg n)$

Δέντρα Διαστημάτων

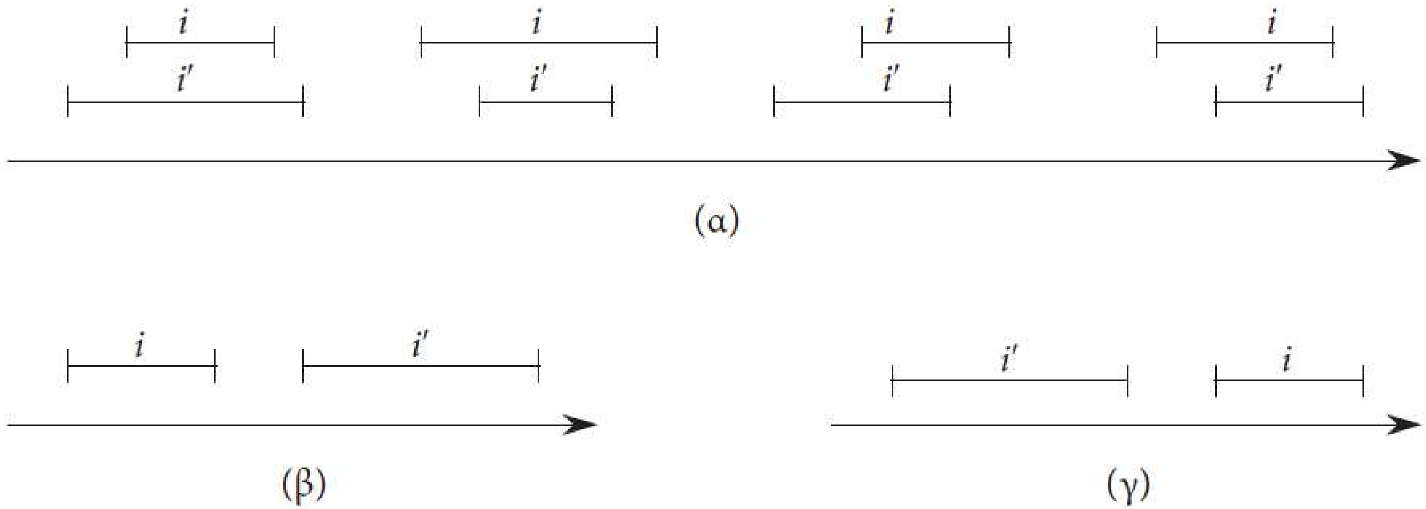
Δέντρα Διαστημάτων (Interval Trees)

- Επαύξηση μελανέρυθρων δέντρων έτσι ώστε να υποστηρίζουν λειτουργίες σε σύνολα διαστημάτων
- Κλειστό διάστημα: ένα διατεταγμένο ζεύγος αριθμών $[t_1, t_2]$, όπου $t_1 \leq t_2$
- Το διάστημα $[t_1, t_2]$ αναπαριστά το σύνολο $\{t \in \mathbb{R} : t_1 \leq t \leq t_2\}$

Δέντρα Διαστημάτων – Αναπαράσταση Διαστήματος

- Αναπαριστούμε ένα διάστημα $[t_1, t_2]$ ως ένα αντικείμενο i, με τα εξής πεδία:
  - $i.low$: t_1 το κάτω άκρο του διαστήματος
  - $i.high$: t_2 το άνω άκρο του διαστήματος
- Επικάλυψη διαστημάτων: δύο διαστήματα i, i' **επικαλύπτονται** αν $i \cap i' \neq \emptyset$ ή αλλιώς αν $i.low \leq i'.high$ και $i'.low \leq i.high$

Δέντρα Διαστημάτων (Τριχοτομία Διαστημάτων)



Τρείς περιπτώσεις για δυο διαστήματα:

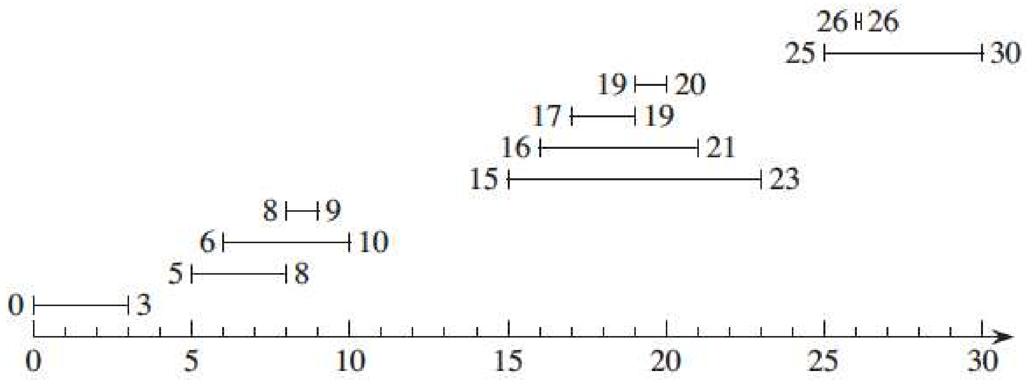
(α) Επικαλυπτόμενα διαστήματα: $i.low \leq i'.high$ και $i'.low \leq i.high$.

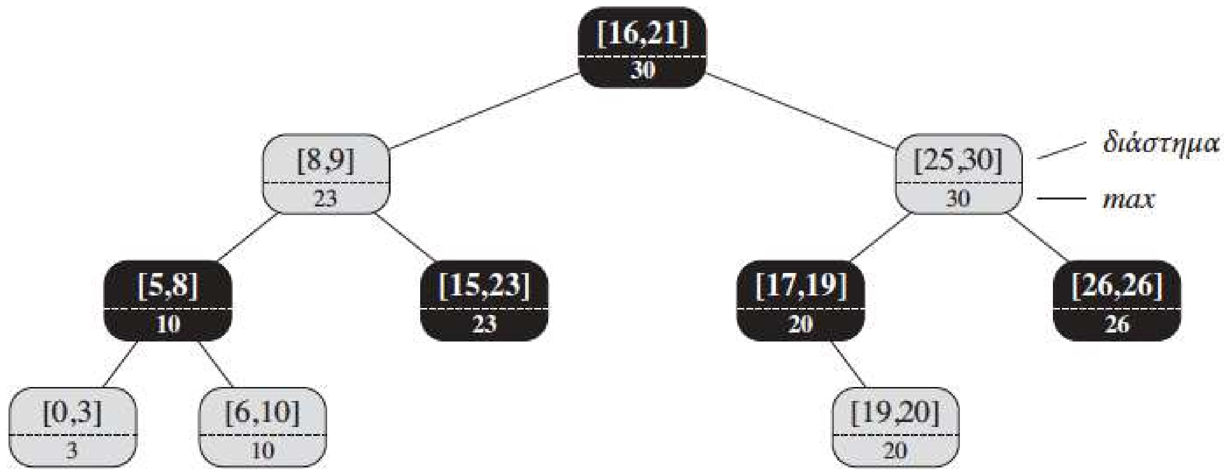
(β)(γ) μη επικαλυπτόμενα διαστήματα: $i.high < i'.low$ ή $i'.high < i.low$

Δέντρα Διαστημάτων

- Ένας κόμβος x του δέντρου πέρα από τους δύο δείκτες στο δεξί και αριστερό παιδί, περιέχει τα εξής πεδία:
  - $x.int$: το διάστημα
  - $x.max$: το μέγιστο άνω άκρο των διαστημάτων των κόμβων που βρίσκονται στο υποδέντρο με ρίζα το x. Ισχύει:
    - $x.max = \max(x.int.high, x.left.max, x.right.max)$
- Κλειδί του κόμβου x: το κάτω άκρο του διαστήματος $x.int.low$

Δέντρα Διαστημάτων - Παράδειγμα





Δέντρα Διαστημάτων

- Λειτουργίες δέντρου διαστημάτων:
  - Interval-Insert(T, x): προσθήκη στοιχείου x (που περιέχει το νέο διάστημα) στο δέντρο T.
  - Interval-Delete(T, x): διαγραφή του στοιχείου x από το δέντρο T
  - Interval-Search(T, i): επιστρέφει δείκτη προς ένα στοιχείο x του δέντρου T, τέτοιο ώστε το $x.int$ να επικαλύπτεται με το διάστημα i, αλλιώς ένα κενό δείκτη

Δέντρα Διαστημάτων - Ενημέρωση

- Εύκολη η διατήρηση πληροφορίας μετά από μία εισαγωγή ή διαγραφή.
- Γνωρίζουμε ότι το $x.max$ εξαρτάται από το $x.int.high, x.left.max, x.right.max$
- Η ενημέρωση μετά από
  – μία εισαγωγή
  – διαγραφή
  – ή περιστροφή (αν απαιτείται)

  γίνεται με ανάλογο τρόπο όπως και στα δέντρα διατακτικής στατιστικής.

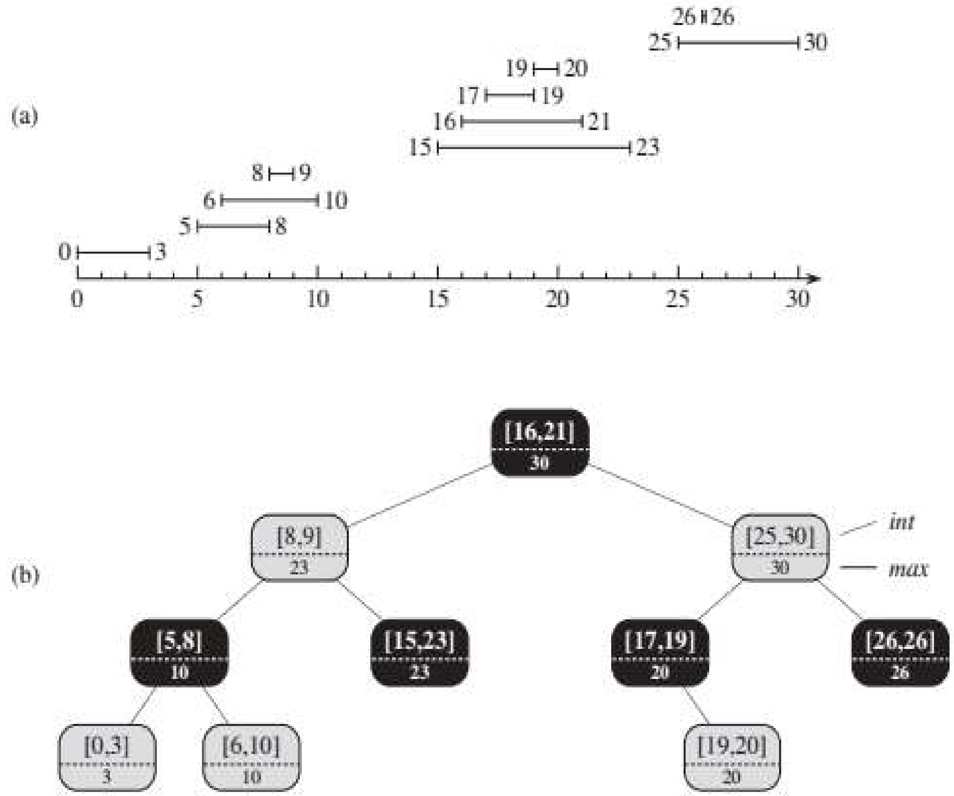
Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

Interval-Search(T, i)

$x = T.root$
while $x \neq T.nil$ **and** i does not overlap $x.int$
    if $x.left \neq T.nil$ **and** $x.left.max \geq i.low$
        $x = x.left$
    else
        $x = x.right$
return x

Δέντρα Διαστημάτων – Αναζήτηση Κόμβου Παράδειγμα

- Αναζήτηση διαστήματος $i = [22, 25]$
  - Εύρεση κόμβου $[15,23]$
- Αναζήτηση διαστήματος $i = [11, 14]$
  - Ανεπιτυχής αναζήτηση



Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

Θεώρημα: Κάθε κλήση της Interval-Search(T, i) επιστρέφει ένα κόμβο του οποίου το διάστημα επικαλύπτεται με το διάστημα i ή επιστρέφει ένα κενό δείκτη όταν το δέντρο T δεν περιέχει διάστημα που να επικαλύπτεται με το i.

Απόδειξη:

- Ο βρόχος στις γραμμές 2-6 τερματίζει όταν $x = T.nil$ ή όταν το διάστημα i επικαλύπτεται με το διάστημα $x.int$.

α) Αν το δέντρο T δεν έχει κανένα διάστημα που να επικαλύπτεται με το διάστημα i:

  - αφού η δεύτερη συνθήκη του while ελέγχει πάντα για επικάλυψη
  - ο αλγόριθμος σωστά θα τερματίσει μόνο όταν προσπαθήσει να πάει πέραν των φύλλων του δέντρου και τότε $x = T.nil$.

Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

β) Έστω ότι τώρα υπάρχει ένα διάστημα στο δέντρο T το οποίο επικαλύπτεται με το διάστημα i .

- Θα αποδείξουμε ότι ισχύει το ακόλουθο πριν από κάθε επανάληψη του while (αναλλοίωτη συνθήκη) :
  - Αν το δέντρο T περιέχει ένα διάστημα που επικαλύπτεται με το διάστημα i , τότε το υποδέντρο με ρίζα το x περιέχει ένα τέτοιο κόμβο.

Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

Αναλλοίωτη συνθήκη: Αν το δέντρο T περιέχει ένα διάστημα που επικαλύπτεται με το διάστημα i, τότε το υποδέντρο με ρίζα το x περιέχει ένα τέτοιο κόμβο

Αρχικός έλεγχος: Στην γραμμή 1 του αλγορίθμου θέτουμε $x \leftarrow T.root$, άρα η συνθήκη ισχύει πριν την πρώτη επανάληψη του while.

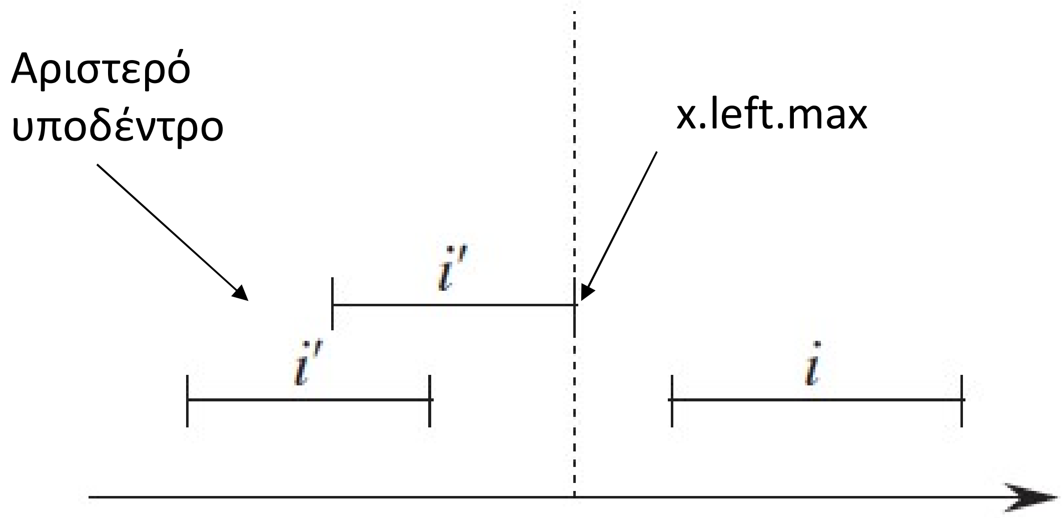
Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

Αναλλοίωτη συνθήκη: Αν το δέντρο T περιέχει ένα διάστημα που επικαλύπτεται με το διάστημα i, τότε το υποδέντρο με ρίζα το x περιέχει ένα τέτοιο κόμβο

Έλεγχος Διατήρησης:

- Σε κάθε επανάληψη του βρόχου εκτελείται είτε η γραμμή 4 είτε η γραμμή 6.
- Θα δείξουμε ότι και στις δύο περιπτώσεις η συνθήκη διατηρείται.
- Έστω ότι εκτελείται η γραμμή 6:
  - Αν $x.left = T.nil$, στο υποδέντρο με ρίζα το $x.left$ $\nexists$ κάποιο διάστημα που να επικαλύπτεται με το $i \Rightarrow$ άρα η αναλλοίωτη συνθήκη διατηρείται.
  - Αν $x.left \neq T.nil$ και $x.left.max < i.low$, ισχύει ότι:
    - $\forall\ i'$ στο υποδέντρο με ρίζα το $x.left$: $i'.high \leq x.left.max < i.low \Rightarrow$
    - i', i δεν επικαλύπτονται $\Rightarrow$
    - άρα στο αριστερό υποδέντρο $\nexists$ διάστημα που να επικαλύπτεται με το $i \Rightarrow$
    - θέτοντας $x = x.right$ η αναλλλοίωτη συνθήκη διατηρείται

Εκτέλεση της 6ης γραμμής



Το διάστημα δεν επικαλύπτεται με τα διαστήματα στο αριστερό υποδέντρο

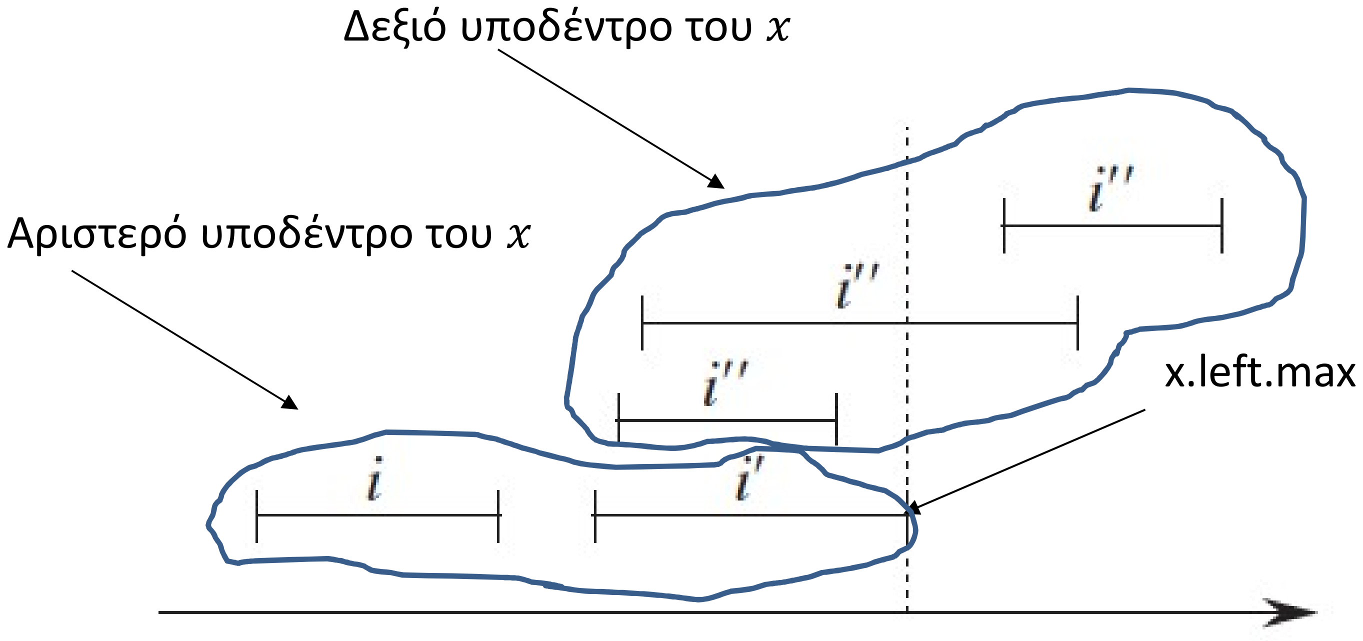
Εκτέλεση της 4ης γραμμής

Αναλλοίωτη συνθήκη:Αν το δέντρο T περιέχει ένα διάστημα που επικαλύπτεται με το διάστημα i, τότε το υποδέντρο με ρίζα το x περιέχει ένα τέτοιο κόμβο

Έλεγχος Διατήρησης (συν.):

- Έστω τώρα ότι εκτελείται η γραμμή 4:
- Θα αποδείξουμε ότι ισχύει η αναλλοίωτη συνθήκη με αντιθετοαντιστροφή (contrapositive):
  - αν το υποδέντρο με ρίζα το $x.left$ δεν περιέχει κάποιο διάστημα που να επικαλύπτεται με το i, τότε κανένα διάστημα στο δέντρο δεν επικαλύπτεται με το i
- Αντιθετοαντιστροφή: μέθοδος απόδειξης.
  - Αν θέλουμε να αποδείξουμε ότι P ⇒ Q αρκεί να αποδείξουμε ότι not Q ⇒ not P
- Ισχύει ότι $x.left.max \geq i.low$
- Επίσης από τον ορισμό του max, το αριστερό υποδέντρο του x πρέπει να περιέχει διάστημα i' τέτοιο ώστε $i'.high = x.left.max \geq i.low$
- Αφού i, i' δεν επικαλύπτονται και επιπλέον $i'.high \geq i.low \Rightarrow i.high < i'.low$ (Τριχοτομία Διαστημάτων).
- Επίσης, το δέντρο διαστημάτων είναι ένα δυαδικό δέντρο αναζήτησης ως προς το κάτω άκρο των διαστημάτων: συνεπάγεται ότι ∀ i'' στο δεξί υποδέντρο του x ισχύει: $i.high < i'.low \leq i''.low \Rightarrow i, i''$ δεν επικαλύπτονται
- Άρα δεν υπάρχει διάστημα στο δεξιό υποδέντρο του x που να επικαλύπτεται με το i.
- Επομένως δεν υπάρχει στο δέντρο T κάποιο διάστημα που να επικαλύπτεται με το i.
- Άρα η αναλλοίωτη συνθήκη αποδείχθηκε με τη μέθοδο της αντιθετοαντιστροφής.

Εκτέλεση της 4ης γραμμής



Το διάστημα i δεν επικαλύπτεται με τα διαστήματα του αριστερού υποδέντρου του κόμβου $x \Rightarrow$ το διάστημα i δεν επικαλύπτεται με τα διαστήματα i'' του δεξιού υποδέντρου.

Δέντρα Διαστημάτων – Αναζήτηση Κόμβου

Αναλλοίωτη συνθήκη:Αν το δέντρο T περιέχει ένα διάστημα που επικαλύπτεται με το διάστημα i, τότε το υποδέντρο με ρίζα το x περιέχει ένα τέτοιο κόμβο

Άρα αποδείξαμε με επαγωγή ότι η αναλλοίωτη συνθήκη ισχύει πριν από κάθε επανάληψη του βρόχου while.

Επομένως ισχύει και πριν την τελευταία όπου ο έλεγχος της συνθήκης του while απέτυχε.

Αφού ισχύει η αναλλοίωτη συνθήκη και αφού εξετάζουμε την περίπτωση που το T περιέχει διάστημα επικαλυπτόμενο με το διάστημα i, ο βρόχος τερματίζει γιατί βρέθηκε το επικαλυπτόμενο διάστημα.