Κόκκινα-Μαύρα Δένδρα (Red-Black Trees)

Ένα κόκκινο-μαύρο δένδρο είναι ένα δυαδικό δένδρο αναζήτησης στο οποίο οι κόμβοι μπορούν να χαρακτηρίζονται από ένα εκ των δύο χρωμάτων: μαύρο-κόκκινο.

Το χρώμα της ρίζας είναι πάντα μαύρο. Αν κάποιο παιδί κάποιου κόμβου δεν υπάρχει, ο αντίστοιχος δείκτης είναι NIL. Θα θεωρήσουμε αυτούς τους δείκτες NIL ως δείκτες σε εξωτερικούς NIL κόμβους (φύλλα) του δυαδικού δένδρου και τους κανονικούς κόμβους που αποθηκεύουν κλειδιά ως εσωτερικούς κόμβους του δένδρου.

Ένα κόκκινο-μαύρο δένδρο πληροί τις ακόλουθες ιδιότητες:

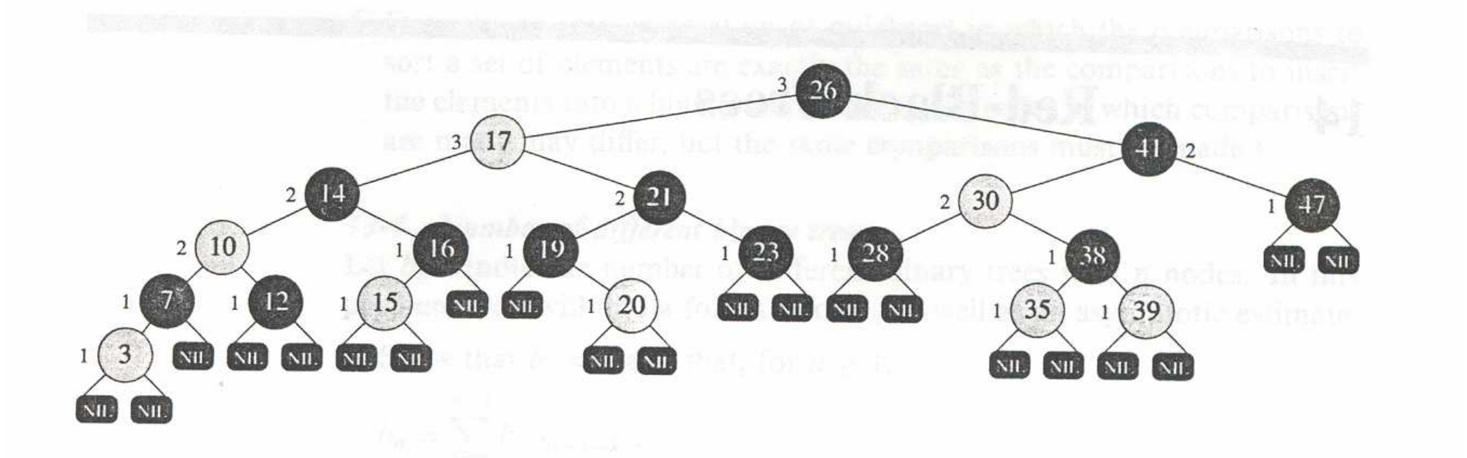
- Κάθε κόμβος έχει είτε κόκκινο είτε μαύρο χρώμα.
- Κάθε NIL κόμβος έχει μαύρο χρώμα.
- Κάθε απλή διαδρομή από έναν κόμβο σε έναν NIL κόμβο που είναι απόγονός του περιέχει το ίδιο πλήθος μαύρων κόμβων.
- Αν ένας κόμβος έχει κόκκινο χρώμα, τότε και τα δύο παιδιά του έχουν μαύρο χρώμα.

Μέγεθος Κόκκινων-Μαύρων Δένδρων

Πρόταση: Ένα κόκκινο μαύρο δένδρο με n εσωτερικούς κόμβους έχει ύψος το πολύ 2log(n+1).

Διαίσθηση: Τουλάχιστον οι μισοί κόμβοι σε κάθε μονοπάτι από τη ρίζα σε φύλλο είναι μαύροι. *Γιατί;*

Απόδειξη



- ✓ Ορίζουμε το μαύρο ύψος ενός κόμβου v, bh(v), να είναι ο αριθμός των μαύρων κόμβων σε κάθε μονοπάτι από τον κόμβο v σε οποιοδήποτε φύλλο, χωρίς να συμπεριλαμβάνουμε τον v.
- ✓ Αποδεικνύουμε ότι το υπο-δένδρο με ρίζα κάποιο κόμβο v περιέχει τουλάχιστον $2^{bh(v)}$-1 εσωτερικούς κόμβους με επαγωγή ως προς το ύψος του v. Στη βάση επαγωγής, αν ο v έχει ύψος 0 τότε είναι φύλλο (ο κόμβος φρουρός), οπότε ο αριθμός των εσωτερικών κόμβων στο υποδένδρο που εκφύεται από τον v είναι 0 όπως απαιτείται. Δοθέντος ενός κόμβου v, υποθέτουμε ότι ο ισχυρισμός ισχύει για όλους τους απογόνους του v και τον αποδεικνύουμε για τον v.
- ✓ Έστω h το ύψος του δένδρου και r η ρίζα. Είναι bh(r) $\geq$ h/2. Άρα, n $\geq 2^{h/2}$ -1 $\Rightarrow$ h $\leq$ 2log(n+1).

Πως υλοποιούμε τη LookUp()? Πολυπλοκότητα?

Εισαγωγή σε Κόκκινο-Μαύρο Δένδρο

Πως κάνουμε εισαγωγή ενός κόμβου x σε κόκκινο-μαύρο δένδρο? Τι πρόβλημα δημιουργείται?

Αλγόριθμος

- Εισαγωγή όπως σε δυαδικό δένδρο αναζήτησης.
- Το μονοπάτι αποθηκεύεται σε στοίβα.
- Ο νέος κόμβος που εισάγεται χαρακτηρίζεται κόκκινος.
- Εξετάζουμε αν οι ιδιότητες χρωματισμού εξακολουθούν να ισχύουν. Αν ναι τερματίζουμε, διαφορετικά διορθώνουμε.

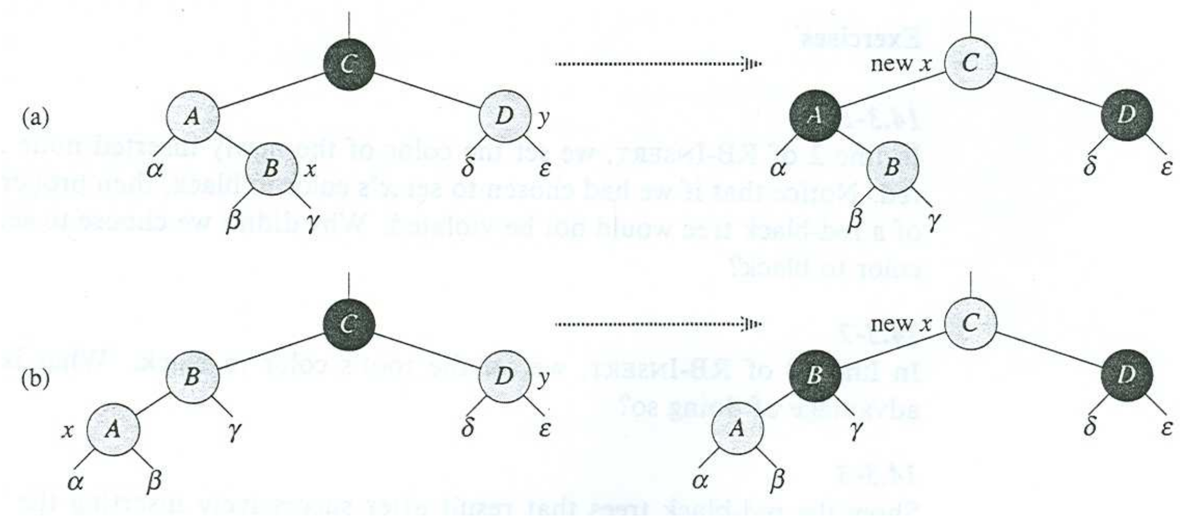
- *Ο πατρικός κόμβος πρέπει να είναι κόκκινος. Γιατί; Ο κόμβος δεν μπορεί να είναι η ρίζα.*
- *Ο παππούς πρέπει να είναι μαύρος. Γιατί?*

Εισαγωγή σε Κόκκινο-Μαύρο Δένδρο

Αλγόριθμος (συνέχεια)

➢ Υποθέτουμε ότι ο πατέρας του x είναι αριστερό παιδί. Διακρίνουμε περιπτώσεις:

1. **Περίπτωση 1:** *Αν ο θείος (αδελφικός κόμβος πατέρα) είναι κόκκινος*, αλλάζουμε το χρώμα του πατέρα και του θείου σε μαύρο και το χρώμα του παππού σε κόκκινο. Επαναλαμβάνουμε.

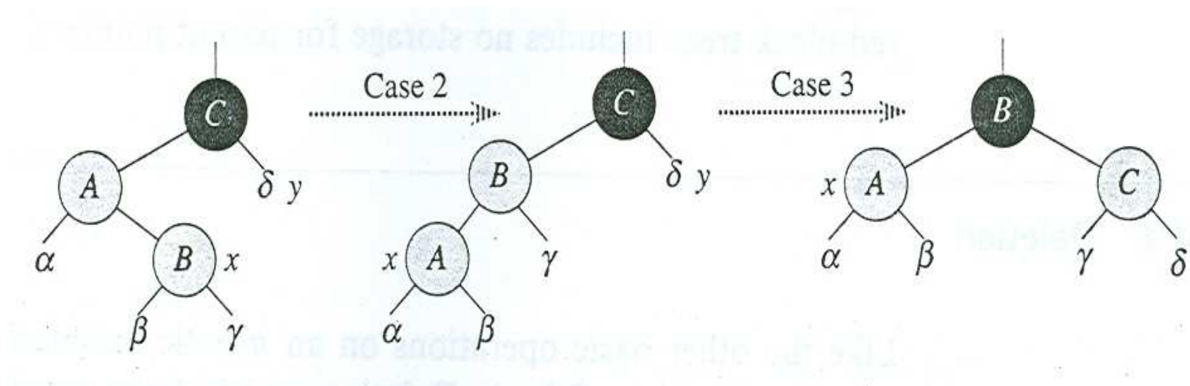


Εισαγωγή σε Κόκκινο-Μαύρο Δένδρο

Αλγόριθμος (συνέχεια)

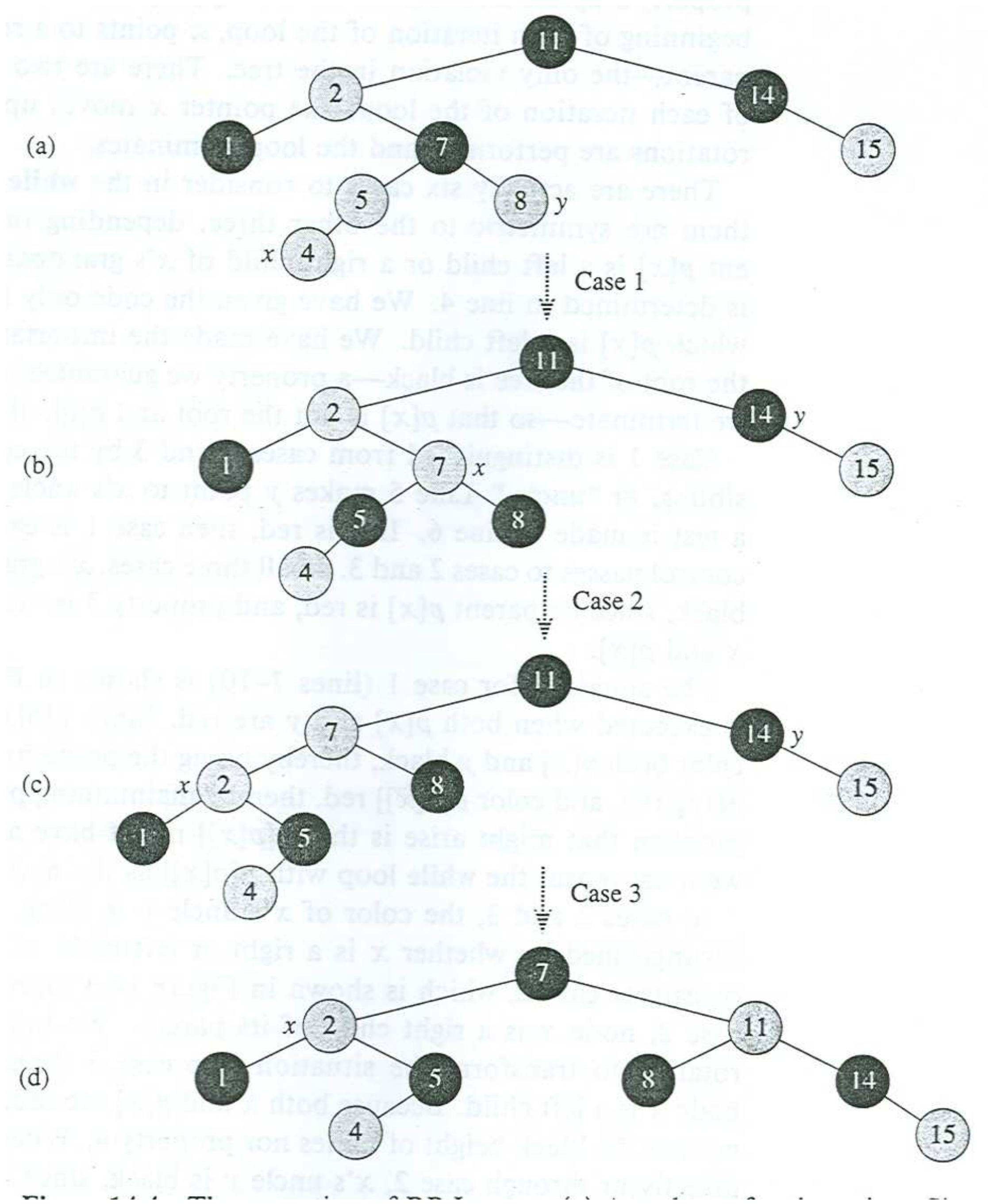
2. **Περίπτωση 2:** Ο θείος είναι μαύρος και ο x είναι δεξί παιδί του πατέρα του. Ανάγουμε την περίπτωση αυτή στην περίπτωση 3 με την εκτέλεση μιας αριστερής περιστροφής.

3. **Περίπτωση 3:** Ο x είναι αριστερό παιδί του πατέρα του. Το χρώμα του πατέρα του x αλλάζει σε μαύρο και του παππού σε κόκκινο. Εκτελείται μια δεξιά περιστροφή.



➤ *Ποια είναι η πολυπλοκότητα της RB-Insert()?*

Εισαγωγή σε Κόκκινο-Μαύρο Δένδρο: Παράδειγμα



Διαγραφή από κόκκινο-μαύρο δένδρο

Η διαγραφή γίνεται με παρόμοιο τρόπο όπως σε δυαδικό δένδρο αναζήτησης και στη συνέχεια ελέγχεται αν οι ιδιότητες χρωματισμού ισχύουν ή όχι και γίνονται κατάλληλες ενέργειες.

Έστω y ο κόμβος που θα διαγραφεί από το δένδρο.

Αν y κόκκινος, δεν υπάρχει πρόβλημα. *Γιατί*?

Αν y μαύρος, η διαγραφή του δημιουργεί (τουλάχιστον) ένα μονοπάτι με μαύρο ύψος μικρότερο κατά ένα.

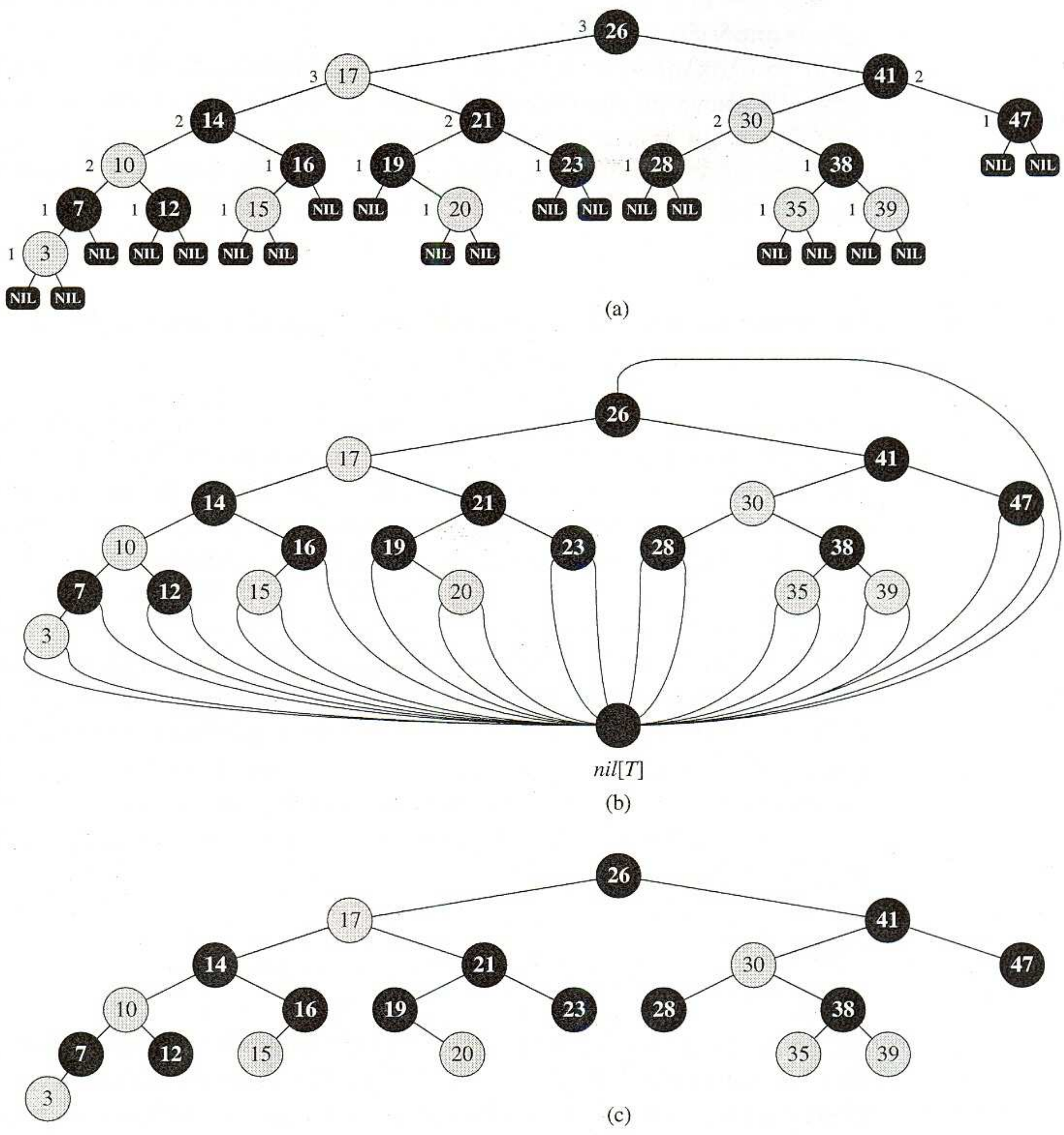
Υποθέτουμε ότι η μαύρη ιδιότητα του v μεταφέρεται στο παιδί του, το οποίο τώρα γίνεται διπλά μαύρο (που είναι μη επιτρεπτό).

Πρέπει να μεταφέρουμε το extra μαύρο προς τα πάνω στο δένδρο μέχρι είτε:

- να φθάσουμε στη ρίζα, ή
- να βρούμε έναν κόκκινο κόμβο που τον μετονομάζουμε σε μαύρο και τερματίζουμε, ή
- να μπορούν να εκτελεστούν κατάλληλες περιστροφές και επαναχρωματισμοί κάποιων κόμβων ώστε να λυθεί το πρόβλημα.

Διαγραφή από κόκκινο-μαύρο δένδρο - Περιπτώσεις

- Θεωρώ ότι το δένδρο υλοποιείται με κόμβο φρουρό. Έτσι, όλοι οι null δείκτες δείχνουν στον κόμβο φρουρό.



- Έστω x το παιδί του κόμβου που διαγράφεται.

Διαγραφή από κόκκινο-μαύρο δένδρο - Περιπτώσεις

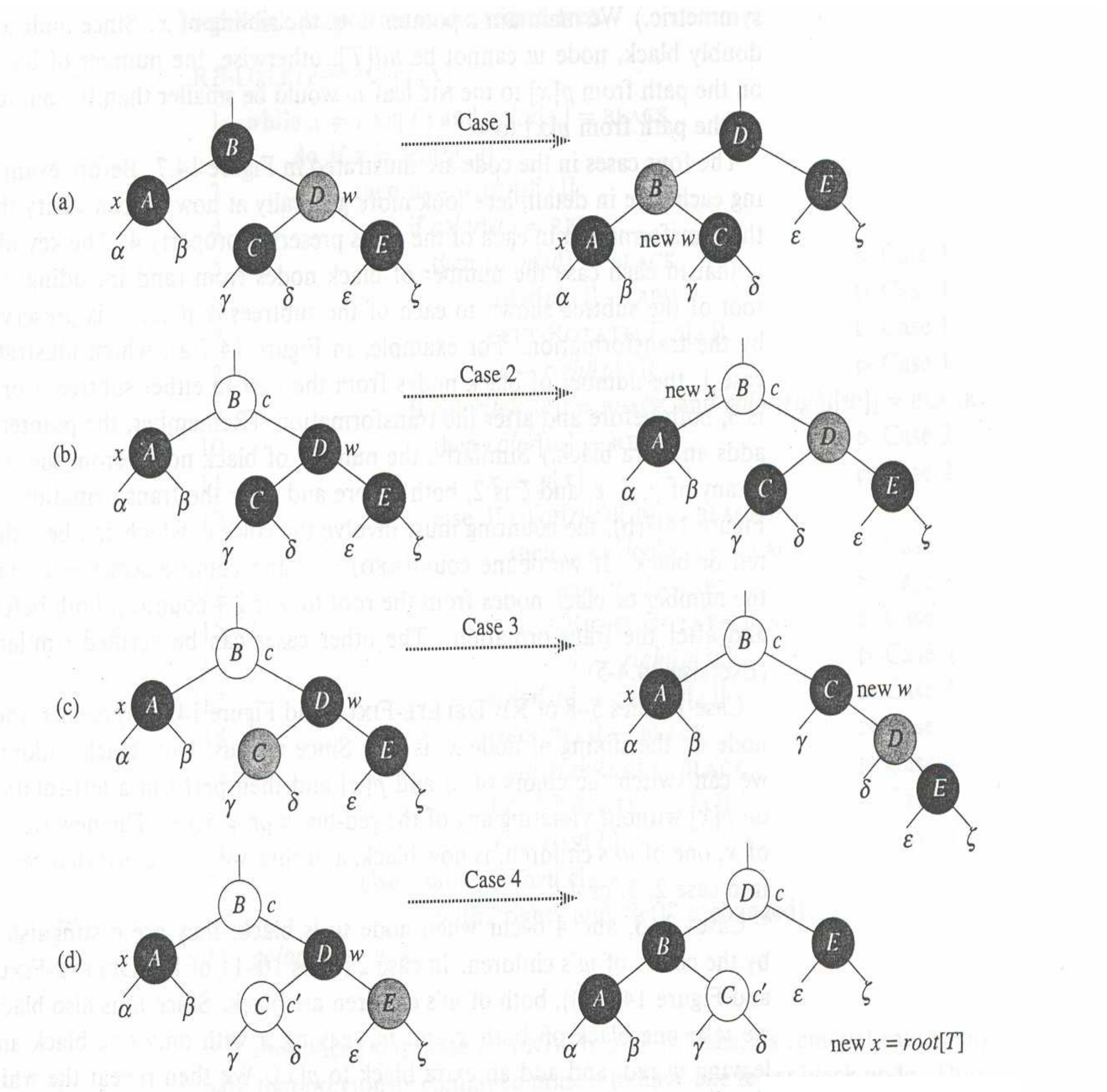
Έστω w ο αδελφικός κόμβος & p ο πατέρας του x. Ο w δεν μπορεί να είναι ο κόμβος φρουρός. *Γιατί;*

Υποθέτουμε ότι ο x είναι αριστερό παιδί του πατρικού του κόμβου. Η περίπτωση που ο x είναι δεξιό παιδί του πατρικού κόμβου είναι συμμετρική.

Διακρίνουμε περιπτώσεις ως προς το χρώμα του w.

1. Ο w είναι κόκκινος. Αλλάζουμε το χρώμα του w σε μαύρο και του p σε κόκκινο και εκτελούμε μια αριστερή περιστροφή γύρω από τον πατέρα του x (περίπτωση α. σχήματος). Έτσι, μεταπίπτουμε στην περίπτωση 2.
2. Ο w είναι μαύρος κόμβος.
    a. Και τα δύο παιδιά του w είναι μαύρα. Αλλάζουμε το χρώμα του w σε κόκκινο, του x σε μαύρο (από διπλά μαύρο) και μεταφέρουμε το μαύρο που αφαιρέσαμε από τους w, x στον p. Αν ο p ήταν κόκκινος γίνεται μαύρος και ο αλγόριθμος τερματίζει. Διαφορετικά, ο p γίνεται διπλά μαύρος και ο αλγόριθμος επαναλαμβάνεται με x = p.
    b. Το w->lc είναι κόκκινο και το w->rc μαύρο. Αλλάζω το χρώμα του w σε κόκκινο και του w->lc σε μαύρο και εκτελώ μια περιστροφή γύρω από το w ⇒ Μεταπίπτουμε στην περίπτωση 2c.
    c. Το w->rc είναι κόκκινο. Αλλάζω το χρώμα του w->rc σε μαύρο, του w σε ότι ήταν το χρώμα του p και του p σε μαύρο και εκτελώ μια περιστροφή γύρω από το p. Ο αλγόριθμος τερματίζει.

Διαγραφή από κόκκινο-μαύρο δένδρο - Περιπτώσεις



Ποια είναι η πολυπλοκότητα της RB-Delete()?