B-Δένδρα

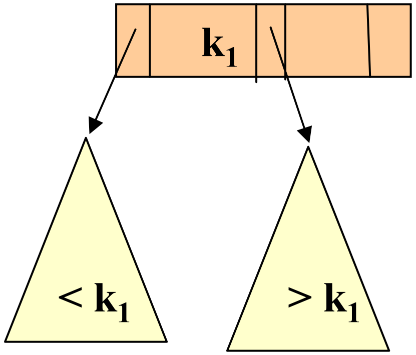
Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

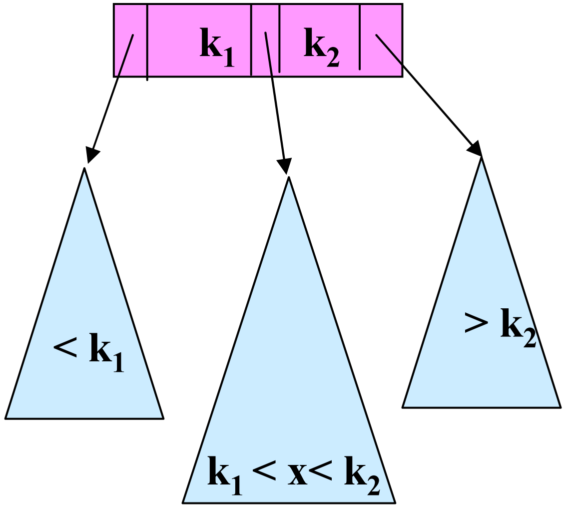
2-3 Δένδρα, Υλοποίηση και πράξεις

B-δένδρα

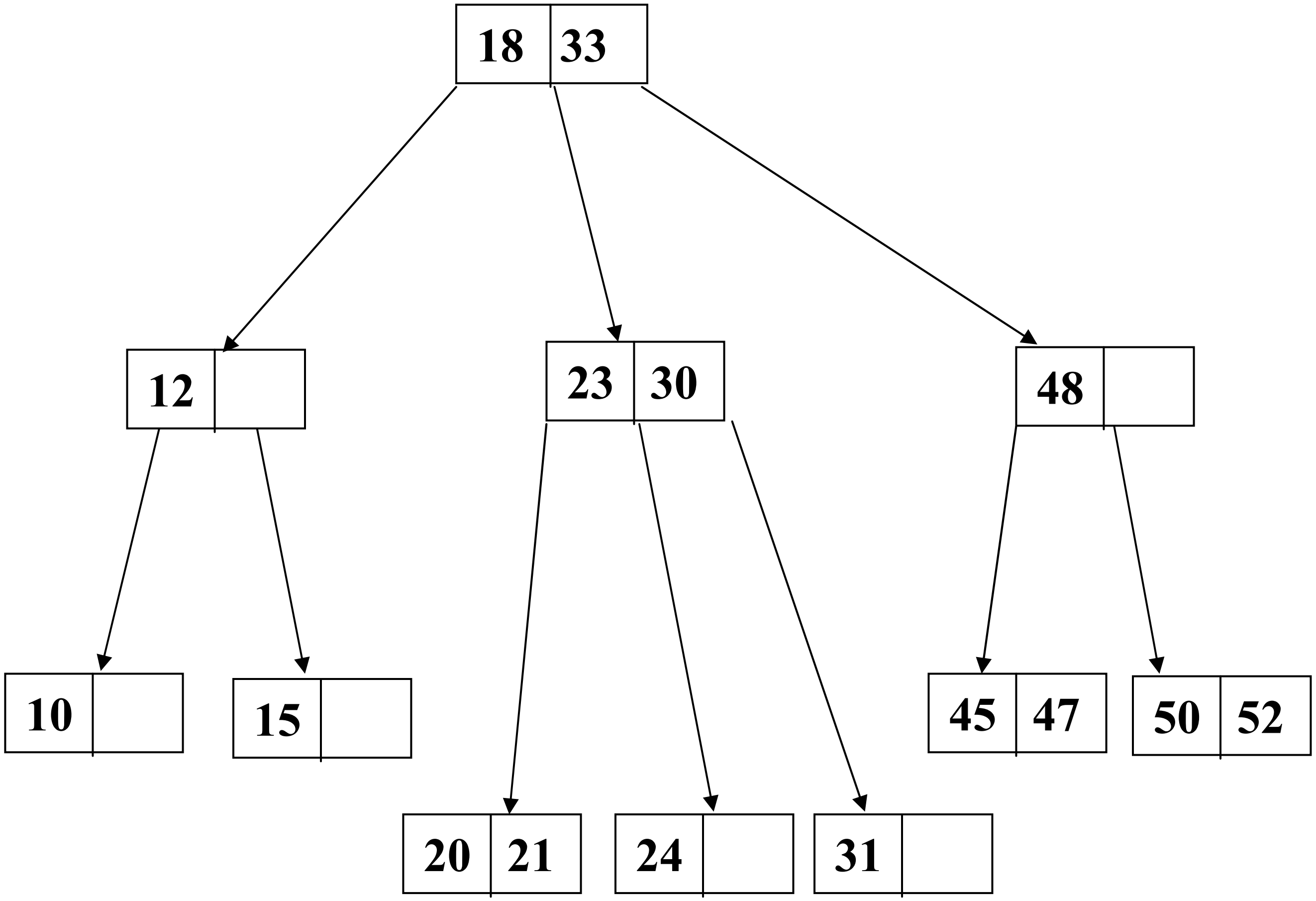
2-3 Δένδρα

- Γενίκευση των δυαδικών δένδρων αναζήτησης.
- Ορισμός:
  - Κάθε κόμβος περιέχει ένα ή δύο κλειδιά.
  - Ένας εσωτερικός κόμβος u με ένα κλειδί , k_1 έχει δύο παιδιά (υπόδενδρα): το αριστερό, u.left, το οποίο περιέχει κλειδιά < k_1, και το μεσαίο, u.center, το οποίο περιέχει κλειδιά > k_1.
  - Ένας εσωτερικός κόμβος u με δύο κλειδιά, $k_1 < k_2$, έχει τρία παιδιά, το αριστερό, u.left, το μεσαίο, u.center, και το δεξί, u.right. Όλα τα κλειδιά του υποδένδρου u.left είναι < k_1, όλα τα κλειδιά του u.center είναι > k_1 και < k_2 και όλα τα κλειδιά του u.right είναι > k_2.
  - Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.





Παράδειγμα 2-3 Δένδρου



Υλοποίηση 2-3Δένδρων

- Ένας κόμβος 2-3 Δένδρου μπορεί να παρασταθεί ως μια εγγραφή με 6 πεδία:
  1. `numkeys`, τύπου int, που δηλώνει τον αριθμό των κλειδιών που περιέχει ο κόμβος,
  2. `key1`, όπου αποθηκεύεται το πρώτο κλειδί,
  3. `key2`, όπου αποθηκεύεται το δεύτερο κλειδί, αν υπάρχει,
  4. `left`, τύπου δείκτη, που δείχνει στο αριστερό παιδί του κόμβου,
  5. `center`, τύπου δείκτη, που δείχνει στο μεσαίο παιδί του κόμβου,
  6. `right`, τύπου δείκτη, που δείχνει στο δεξί παιδί του κόμβου αν υπάρχει.

- Ένα δένδρο αναπαρίσταται ως ένας δείκτης σε κόμβο 2-3 δένδρου (που δείχνει στη ρίζα) και επιτρέπει τις πράξεις εισαγωγής, διαγραφή και αναζήτησης.

Ιδιότητες 2-3 δένδρου

- Αν N(h) είναι ο μικρότερος αριθμός κλειδιών ενός 2-3 δένδρου ύψους h, τότε

$$N(0) = 1, \quad N(h) = 1 + 2 \cdot N(h-1)$$

- Αν M(h) είναι ο μεγαλύτερος αριθμός κλειδιών ενός 2-3 δένδρου ύψους h, τότε

$$M(0) = 2, \quad M(h) = 2 + 3 \cdot M(h-1)$$

- Ο αριθμός κλειδιών ενός 2-3 δένδρου ύψους h είναι το πολύ $3^{h+1} - 1$ και το λιγότερο $2^{h+1} - 1$.

- *Επομένως, το ύψος ενός 2-3 δένδρου με n κόμβους είναι O(log n).*

- Η διαδικασία εύρεσης στοιχείου σε ένα 2-3 δένδρο είναι εύκολη (παρόμοια με αυτή ενός ΔΔΑ).

  Ο χρόνος εκτέλεσης της είναι τάξης .

Εισαγωγή κόμβου

Για την εισαγωγή κλειδιού k σε ένα 2-3 Δένδρο ακολουθούμε το μονοπάτι από τη ρίζα του δένδρου μέχρι το κατάλληλο φύλλο u. Υπάρχουν δύο περιπτώσεις

1. Αν ο u περιέχει μόνο 1 κλειδί, τότε προσθέτουμε το k στον u.

2. Αν ο u περιέχει 2 κλειδιά έστω k_1 και k_2, τότε δημιουργούμε ένα καινούριο κόμβο v, και διαμοιράζουμε τα κλειδιά του u, και το καινούριο κλειδί k έτσι ώστε: ο u να πάρει το μικρότερο των κλειδιών και ο v το μεγαλύτερο. Ο δείκτης προς τον v και το μεσαίο κλειδί, m, μεταβιβάζεται στον πατέρα w του u.

- Αν ο w έχει μόνο ένα κλειδί, τότε προσθέτουμε το ζεύγος (m,v) στον κόμβο w. Διαφορετικά, δημιουργούμε ένα καινούριο κόμβο και επαναλαμβάνουμε το βήμα 2.
- Αυτή η διαδικασία μπορεί να συνεχιστεί μέχρι τη ρίζα του δένδρου με αποτέλεσμα τη διάσπαση της ρίζας και την αύξηση του ύψους του δένδρου κατά 1.

Αναλυτική Περιγραφή – Φάση 1

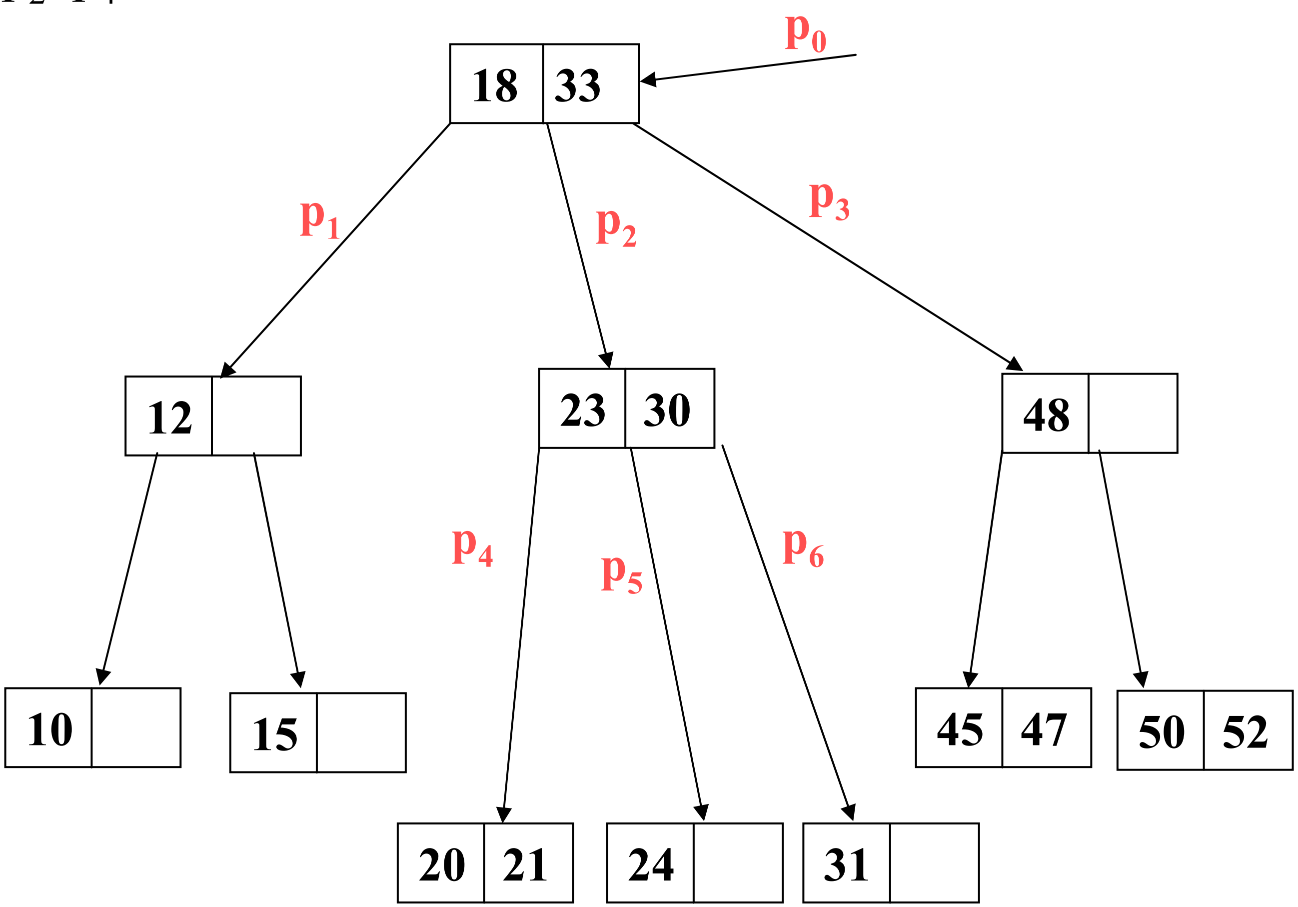
- Ξεκινώντας από τη ρίζα, εφάρμοσε τον πιο κάτω αλγόριθμο σε κάθε κόμβο u μέχρι να φτάσεις σε κάποιο φύλλο, καταγράφοντας τη διαδρομή που ακολούθησες:

- Αν u.numkeys = 1, τότε
  - αν k<u.key1 τότε προχώρα στον κόμβο u.left
  - αν k>u.key1 τότε προχώρα στον κόμβο u.center
  - αν k=u.key1 τότε σταμάτα.

- Αν u.numkeys = 2, τότε
  - αν k<u.key1 τότε προχώρα στον κόμβο u.left
  - αν k>u.key2 τότε προχώρα στον κόμβο u.right
  - αν u.key1 < k < u.key2 τότε προχώρα στον κόμβο u.center.
  - διαφορετικά, σταμάτα.

Τρέχον παράδειγμα – εισαγωγή του 19 (1)

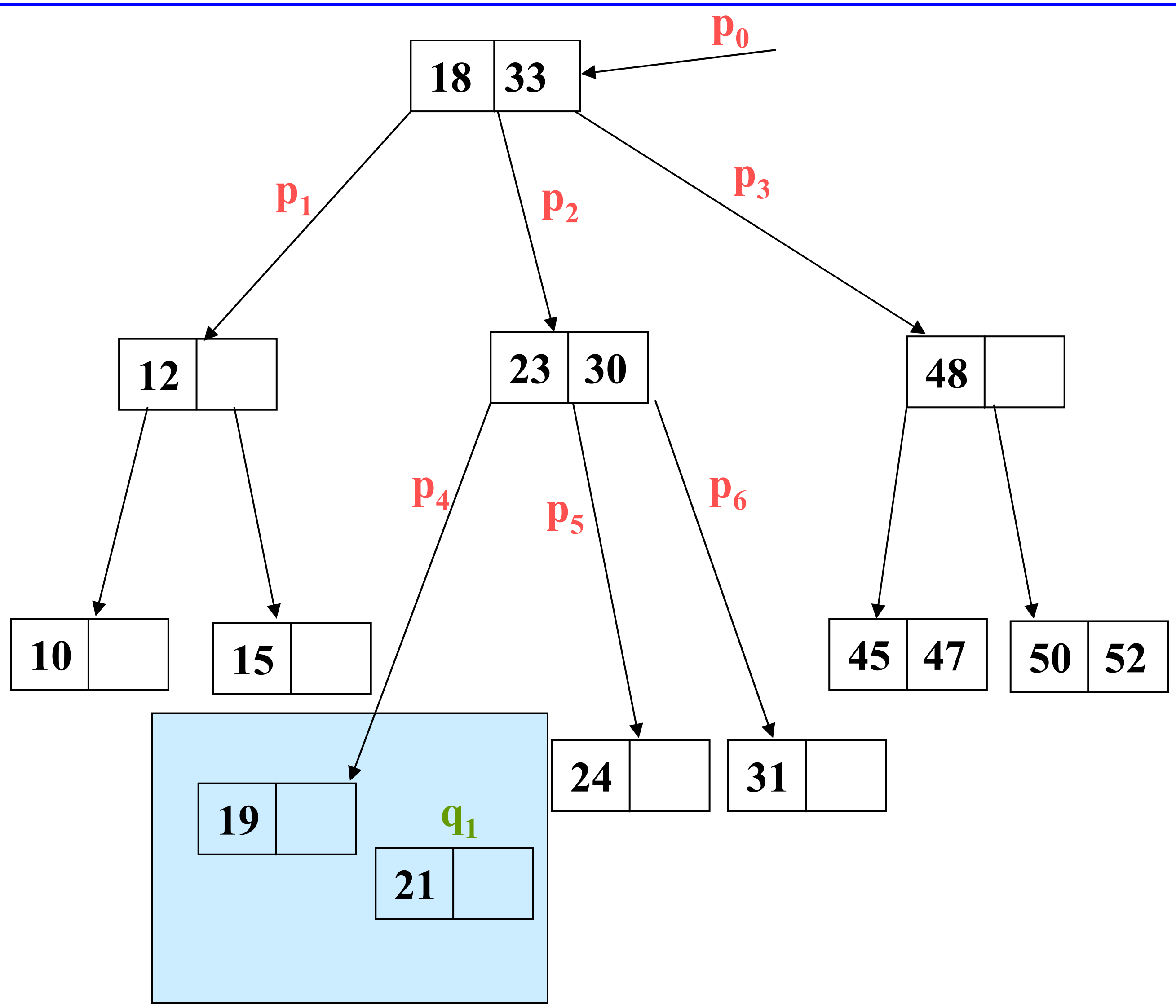
Ξεκινώντας από τη ρίζα, p_0, ακολουθούμε και καταγράφουμε τη διαδρομή p_0, p_2, p_4.



Φάση 2(α) – Περίπτωση Φύλλου

- Έστω ότι προσθέτουμε το κλειδί k στο φύλλο u.
- <u>*Αν u.numkeys = 1*</u>, k_1= u.key1, τότε
  - u.key1 = min (k_1, k),
  - u.key2 = max (k_1, k),
  - u.numkeys = 2
- <u>*Αν u.numkeys = 2*</u>, k_1 = u.key1, k_2 =u.key2, τότε
  - u.key1 = min (k_1, k_2, k),
  - u.numkeys = 1,
  - p = **(NODE *)malloc(sizeof(NODE)),**
  - p.numkeys = 1,
  - p.key1 = max(k_1, k_2 ,k).
  - προχώρα στη φάση 2(β) με παραμέτρους p, mid(k_1, k_2 ,k)

Τρέχον παράδειγμα – εισαγωγή του 19 (2)



Φάση 2(β) – Περίπτωση Εσωτερικού κόμβου

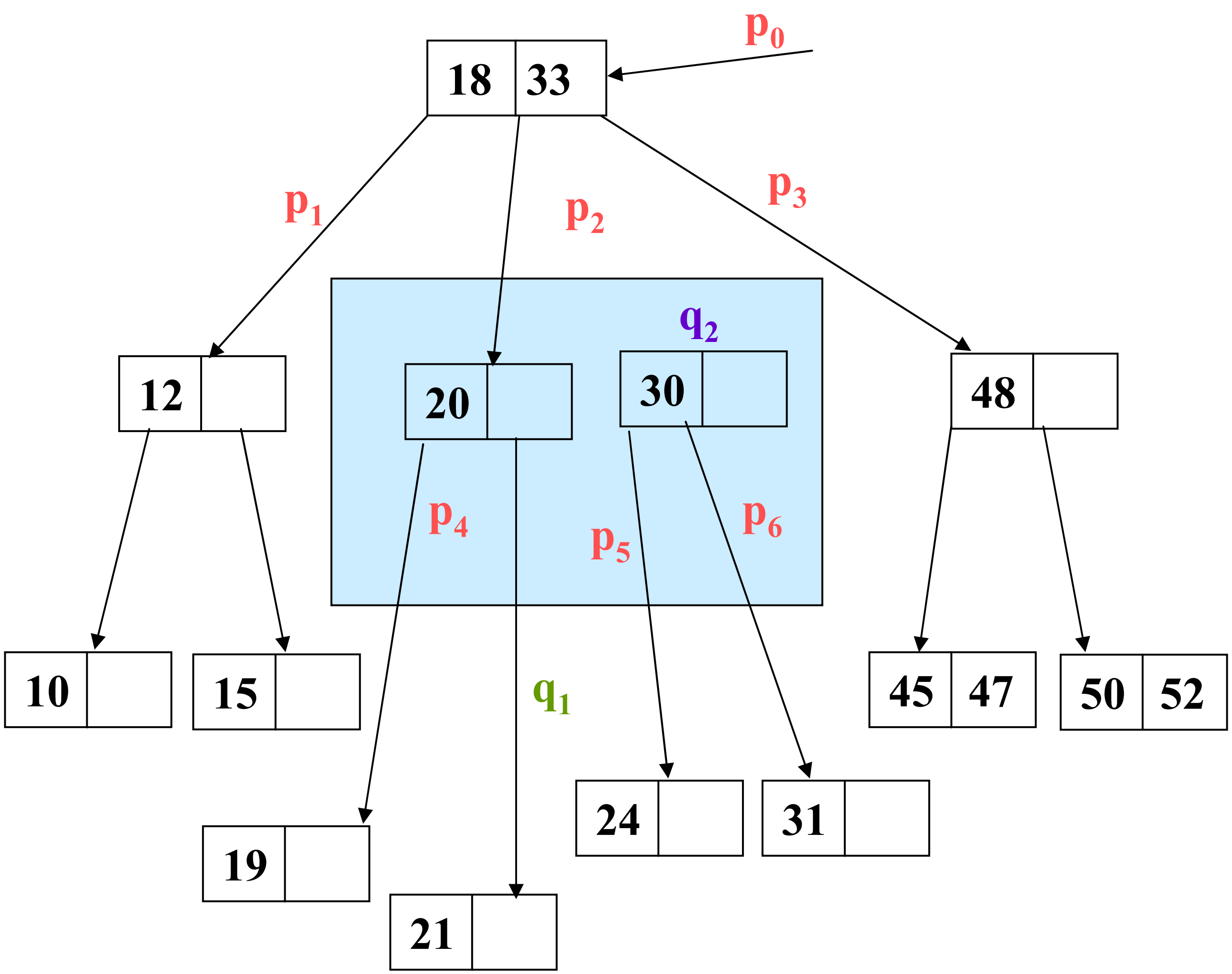
- Έστω ότι προσθέτουμε το κλειδί k και τον δείκτη p στον κόμβο u.

- *Αν u.numkeys = 1*, k_1= u.key1, τότε
  - αν $k > k_1$, `u.key2 = k, u.right = p,`
  - αν $k < k_1$, `u.key1 = k, u.key2 =` k_1`, u.right = u.center, u.center = p.`
  - `u.nunkeys = 2.`

- Αν u.numkeys = 2, k_1= u.key1, k_2= u.key2, u.left = p_1, u.center = p_2 , u.right = p_3, τότε υπάρχουν 3 περιπτώσεις που εξαρτώνται από τη σχέση των κλειδιών k, k_1, k_2. Ας ασχοληθούμε με την περίπτωση $k < k_1$ (οι άλλες δύο είναι παρόμοιες). Τότε
  - `u.key1 = k, u.center = p, u.numkeys = 1,`
  - `q= (NODE *)malloc(sizeof(NODE)), q.numkeys = 1, q.key1 =` k_2`, q.left = p2, q.center = p3.`
  - αν ο u είναι η ρίζα τότε προχώρα στη φάση 3, διαφορετικά επανάλαβε τη φάση 2(β) με παραμέτρους (q, k_1) στον πατέρα του u.

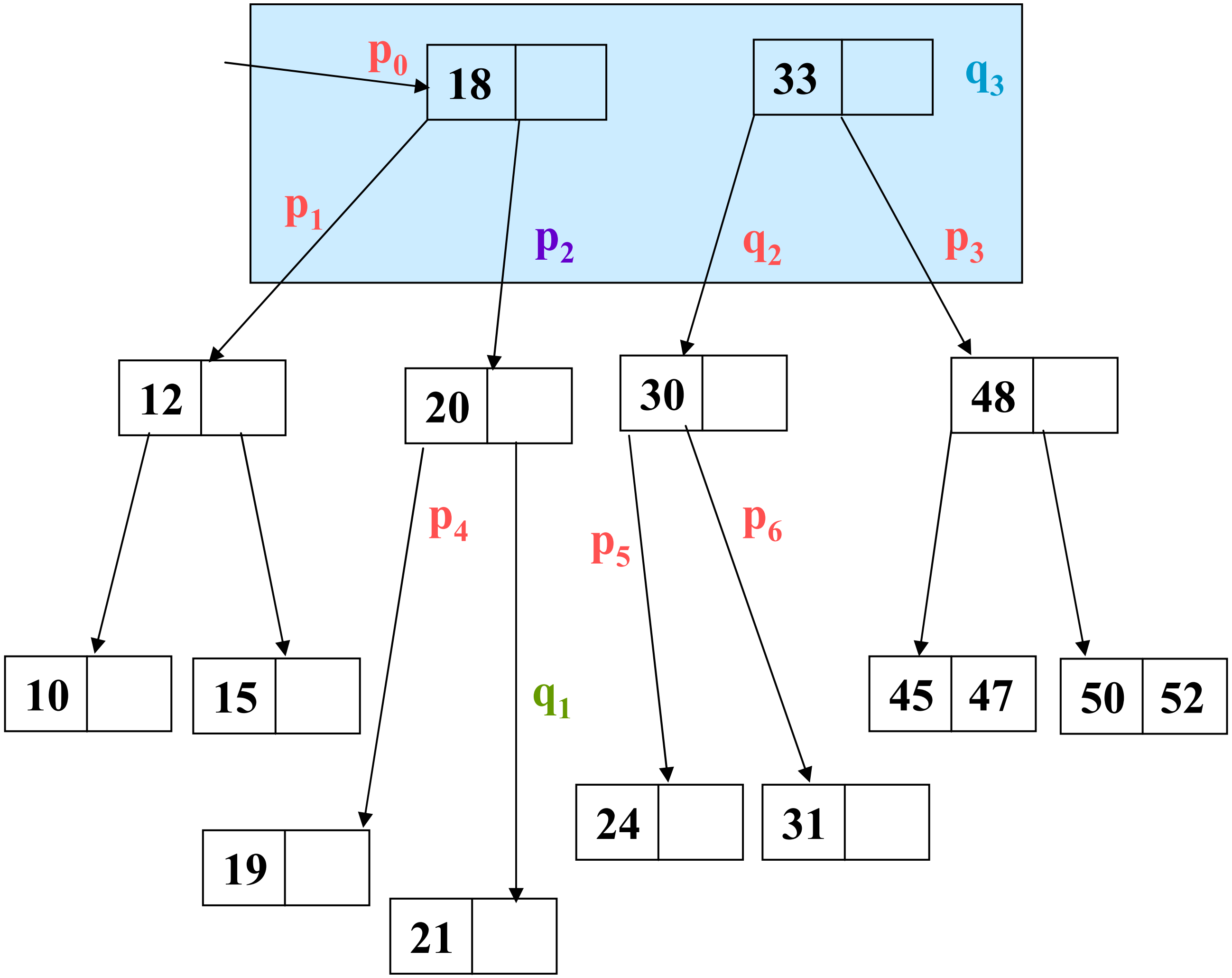
Τρέχον παράδειγμα – εισαγωγή του 19 (3)

- Εισαγωγή του ζεύγους $(20,q_1)$ στον κόμβο p_2.



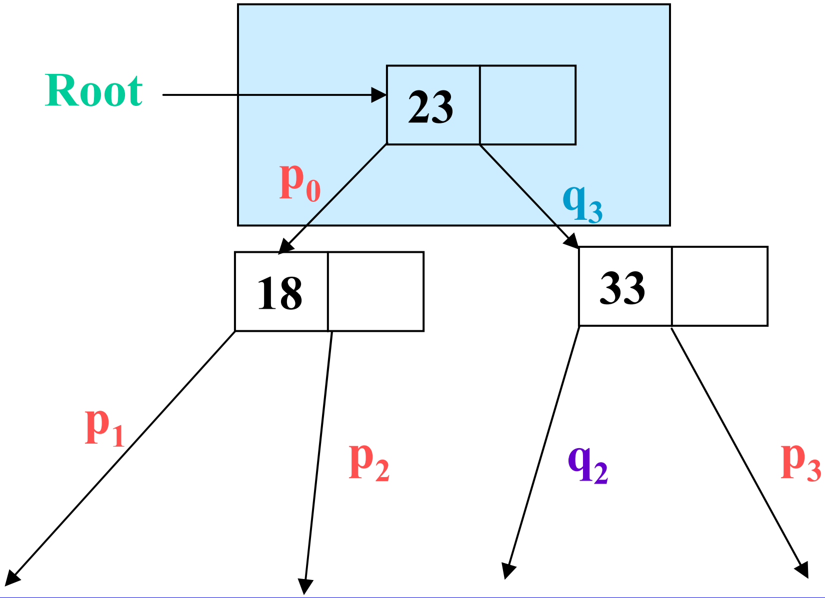
Τρέχον παράδειγμα – εισαγωγή του 19 (4)

- Εισαγωγή του ζεύγους (23,q_2) στον κόμβο p_0.

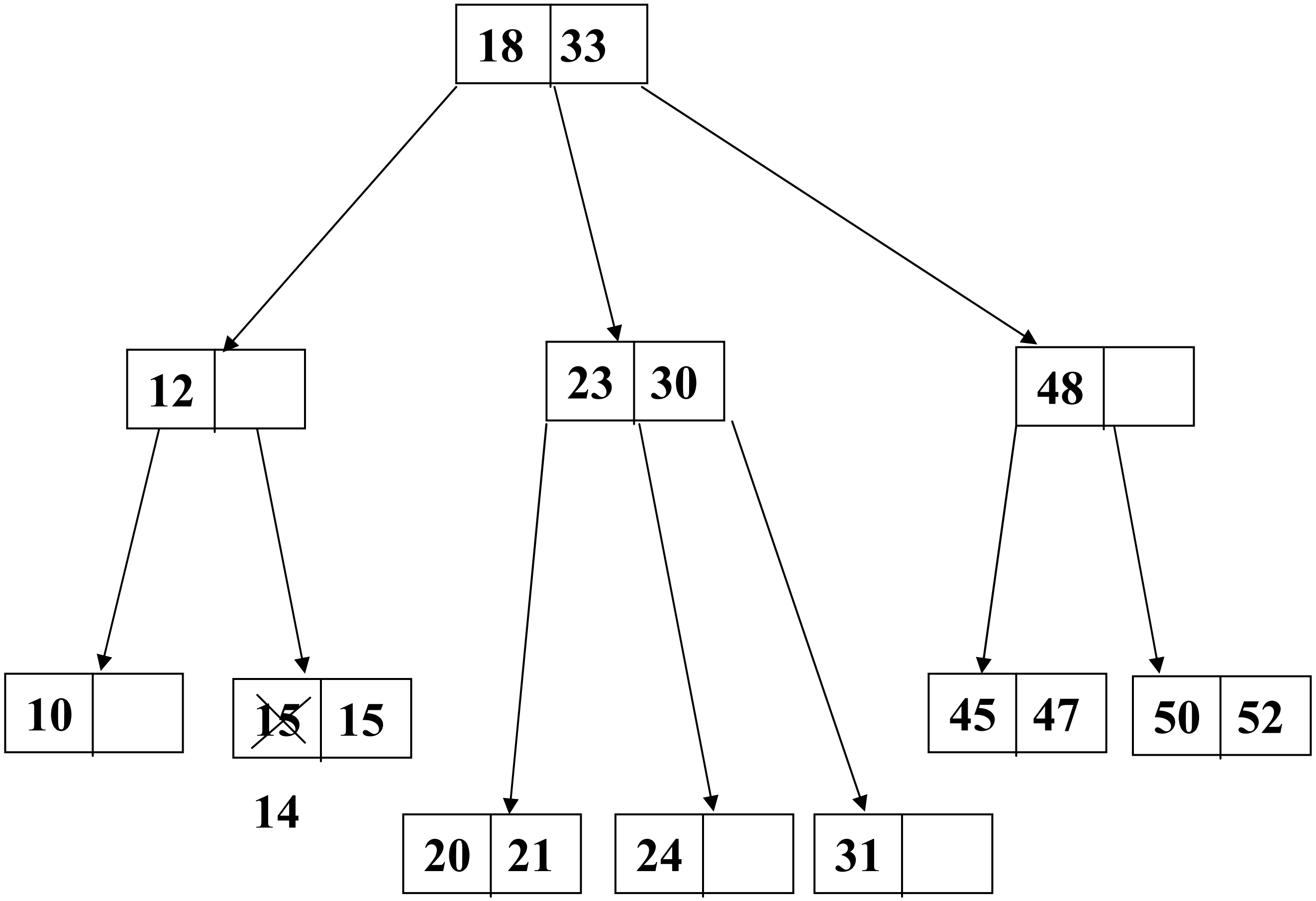


Φάση 3: Νέα ρίζα

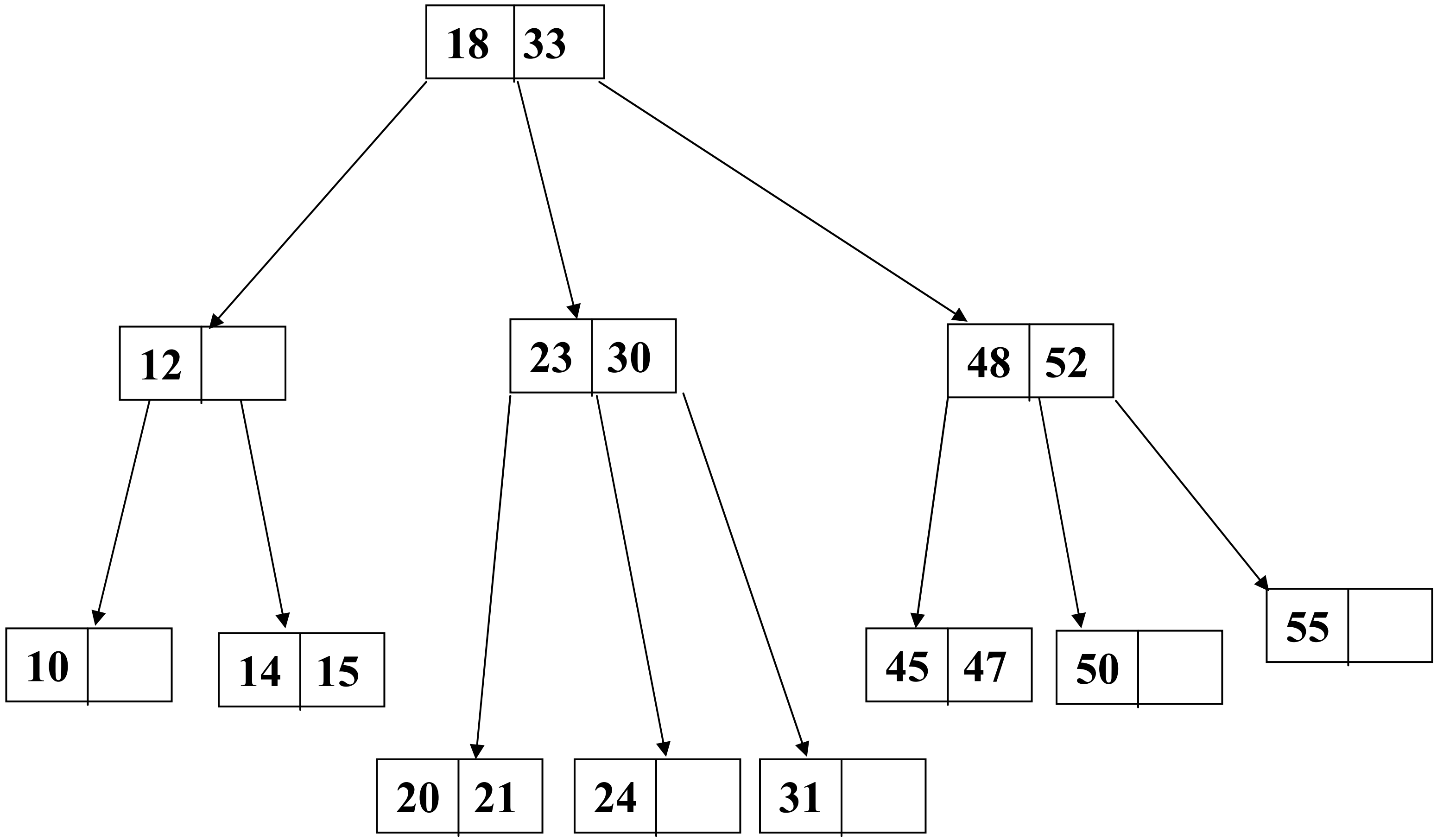
- Αν η ρίζα r διασπαστεί και ζητήσει την εισαγωγή ζεύγους (p,k) στον "ανύπαρκτο" πρόγονο της στο δένδρο, τότε
  - Δημιούργησε ένα καινούριο κόμβο v με ένα κλειδί k και v.left = r, v.center = p.
- Ο καινούριος κόμβος v θα είναι η νέα ρίζα του δένδρου.
- Έτσι, στο παράδειγμα, εισαγωγή του (q_3, 23) έχει σαν αποτέλεσμα:



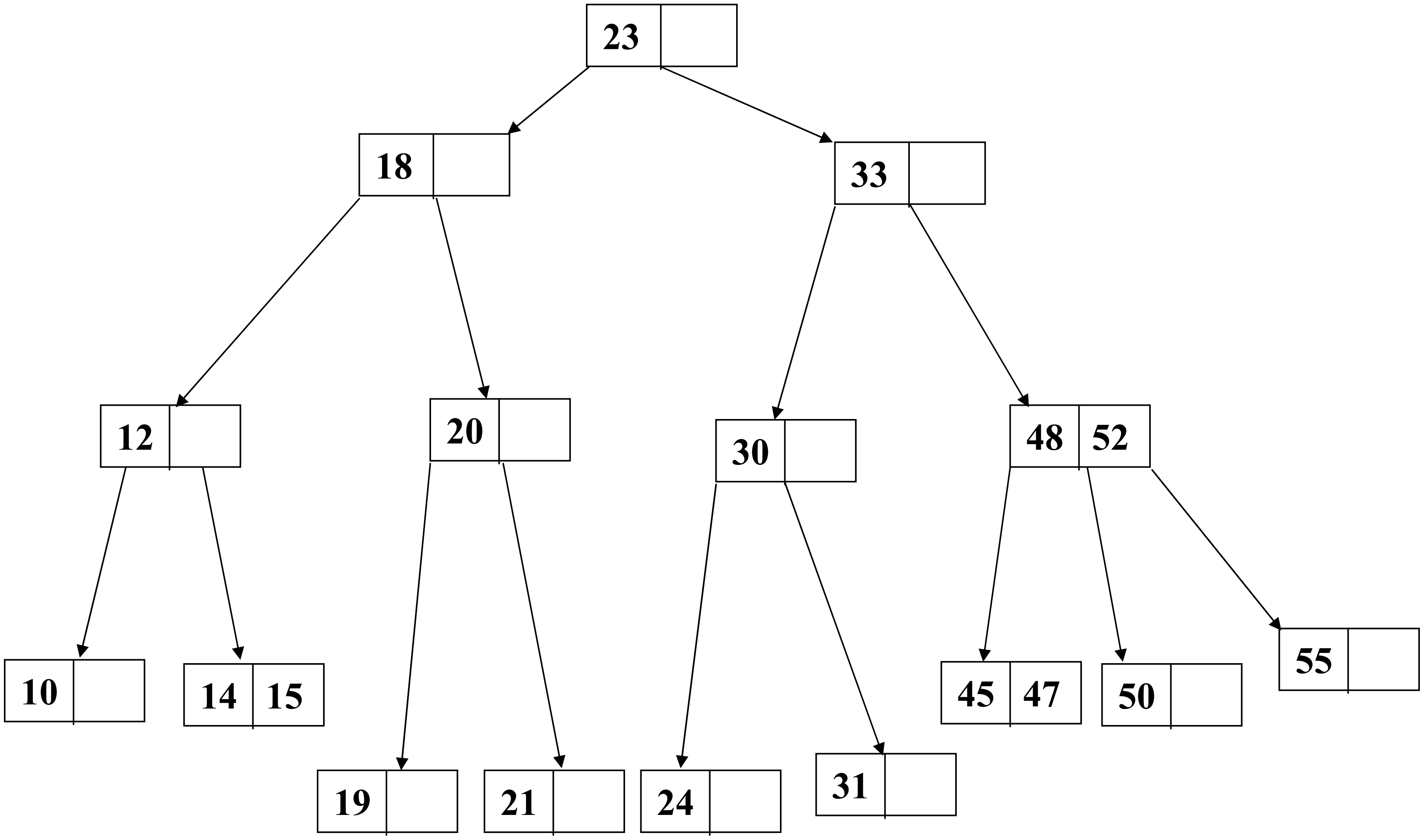
Εισαγωγή του κλειδιού 14



Εισαγωγή του κλειδιού 55



Εισαγωγή του κλειδιού 19

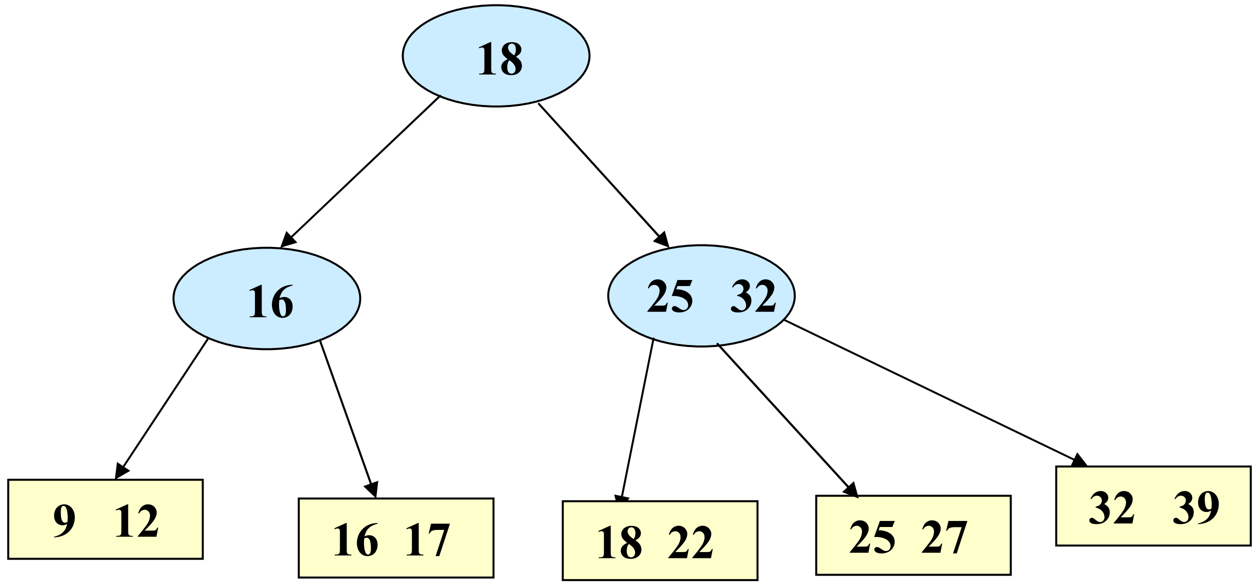


Διαδικασία Εισαγωγής Κόμβου

- Η αναδρομική διαδικασία εισαγωγής κόμβου, παίρνει παραμέτρους:
  1. το δείκτη στον κόμβο όπου θα γίνει η εισαγωγή,
  2. το κλειδί που θέλουμε να προσθέσουμε (κατά την κάθοδο στο δένδρο),
  3. το ζεύγος (δείκτη σε παιδί, κλειδί) που θέλουμε να προσθέσουμε (κατά την άνοδο στο δένδρο), και
- Η διαδικασία εξαγωγής κλειδιών χρησιμοποιεί παρόμοιες ιδέες. Σ'αυτή όμως την περίπτωση, αντί *διάσπαση* κόμβου έχουμε *συγχώνευση* κόμβων.
- Όλες οι διαδικασίες έχουν χρόνο εκτέλεσης ανάλογο με το ύψος του δένδρου. Άρα είναι τάξης Θ(log n).

Παραλλαγή

- Οι πληροφορίες αποθηκεύονται μόνο στα φύλλα.
- Τα κλειδιά εσωτερικών κόμβων χρησιμεύουν μόνο για την αναζήτηση στο δένδρο. Δηλαδή, για κάποιο κόμβο u, το u.key1 δηλώνει το μικρότερο κλειδί σε φύλλο του u.center, και το u.key2 (αν υπάρχει) δηλώνει το μικρότερο κλειδί σε φύλλο του u.right.



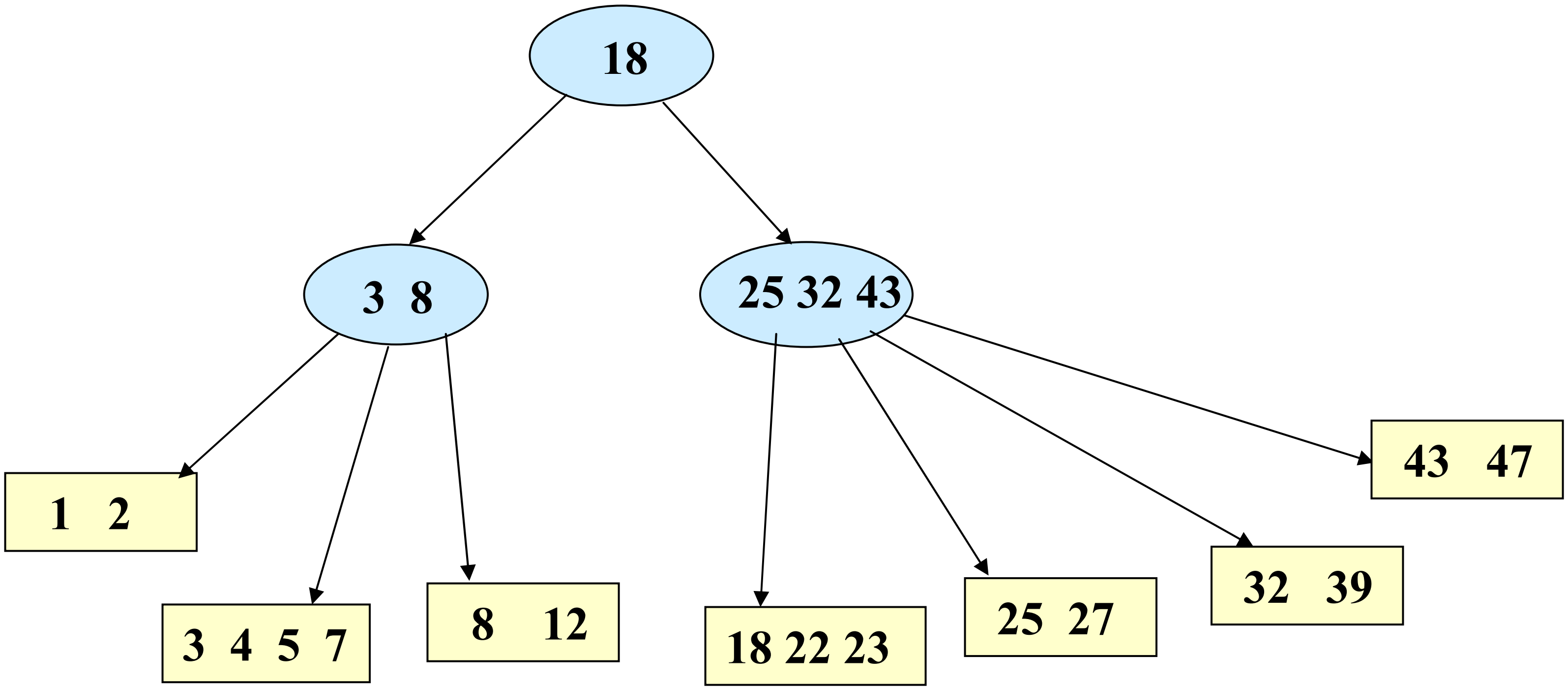
Β-δένδρα

- Γενίκευση του 2-3 δένδρου: μπορεί να αποθηκεύει περισσότερα από 2 κλειδιά σε κάθε κόμβο.

- Ένα δένδρο τάξης m ικανοποιεί τις πιο κάτω ιδιότητες:
  - Έχει μια ρίζα η οποία είτε είναι φύλλο, είτε έχει 2 μέχρι m παιδιά.
  - Όλοι οι εσωτερικοί κόμβοι (εκτός τη ρίζα) έχουν από $\lceil m/2 \rceil$ μέχρι m παιδιά.
  - Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.
  - Τα δεδομένα φυλάσσονται στα φύλλα. Κάθε φύλλο περιέχει από $\lceil m/2 \rceil$ μέχρι m κλειδιά.
  - Ένας εσωτερικός κόμβός με δείκτες p_0, p_1, …, p_n, έχει κλειδιά $k_1,\ldots,k_n$, όπου k_i είναι το μικρότερο κλειδί του υποδένδρου που δείχνεται από τον δείκτη p_i.

- Ένα 2-3 δένδρο, είναι Β-δένδρο, τάξης 3.

Παράδειγμα Β-δένδρου τάξης 4



Ένα B-δένδρο …

- τάξης m με n κλειδιά έχει ύψος $O(\log_m n)$
- Έχει διαδικασίες παρόμοιες με αυτές ενός 2-3 δένδρου, οι οποίες επίσης μπορεί να προκαλέσουν διάσπαση ή συγχώνευση της ρίζας.
- Έχει το πλεονέκτημα έναντι των 2-3 δένδρων …
- Έχει το μειονέκτημα …
- Το κόστος των διαδικασιών είναι ανάλογο του

  (ύψος του δένδρου) × (κόστος επεξεργασίας ενός κόμβου)
- Το κόστος επεξεργασίας ενός κόμβου O(m) μπορεί να γίνει O(log m) αν κάθε κόμβος οργανωθεί ως AVL-δένδρο.
- Σε βάσεις δεδομένων οι κόμβοι αποθηκεύονται σε δευτερεύουσα μνήμη του υπολογιστή. Το κόστος πρόσβασης ενός κόμβου είναι πολύ μεγαλύτερο απ'αυτό της επεξεργασίας του.
- Η τιμή του m επιλέγεται βάσει του πόσοι δείκτες μπορούν να αποθηκευτούν σε ένα καταχώρημα (block) μνήμης.