

Δομές Δεδομένων - ΗΥ240 - 4η Σειρά Ασκήσεων
Ον/νυμο: Αναγνωστάκης Εμμανουήλ
ΑΜ: 3618

Άσκηση 1

α)
Ο τρόπος με τον οποίο μπορούμε να συνενώσουμε τους δύο βωρούς σε χρόνο $O(n_1)$ είναι να επεκτείνουμε το βωρό n_2 κατά n_1 θέσεις και κάθε φορά με τη δημιουργία μιας θέσης τοποθετούμε και ένα στοιχείο του n_1 . Αυτό μπορεί να γίνει είτε στην αρχή είτε στο τέλος αφού ο βωρός είναι μη ταξινομημένος πίνακας άρα δεν έχουμε ταξινόμηση με βάση την αξία των κλειδιών και έχουμε πρόσβαση σε κάθε κελί σε $O(1)$ αφού έχουμε indexing.

β) HeapDelete(pointer j , Heap h) {
 For(i from 0 until j is found)
 Now i stores the position of the element to be deleted.
 After that we need to push all elements to the left to cover the deleted element.
 For(j From i to h_size-2) $A[j] = A[j+1]$;
}

Abkürzung 2

a)

$$S_1 = \{1\}$$

$$S_2 = \{2\}$$

$$S_3 = \{3\}$$

$$S_4 = \{4\}$$

$$S_5 = \{5\}$$

$$S_6 = \{6\}$$

$$S_7 = \{7\}$$

$$S_8 = \{8\}$$

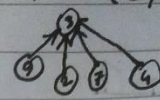
$$S_9 = \{9\}$$

$$S_{10} = \{10\}$$

$$A = \text{Union}(S_3, S_2);$$



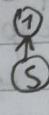
$$D = \text{Union}(C, S_4);$$



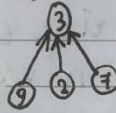
$$B = \text{Union}(S_7, S_9);$$



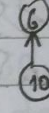
$$E = \text{Union}(S_1, S_5);$$



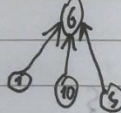
$$C = \text{Union}(A, B);$$



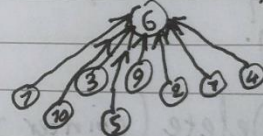
$$F = \text{Union}(S_6, S_{10});$$



$$G = \text{Union}(F, E);$$



$$I = \text{Union}(G, D);$$



β) Για να είναι το directory όσο το δυνατόν μικρότερο πρέπει να ερμηνεύουμε συνεχώς να δείχνουν βέε ένα κλάδο, δηλαδή να ερμηνεύουμε το $b=2$ και να μην δημιουργούμε περιττές βελίδες.

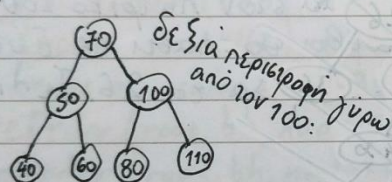
Είσαγωγή του 00100:

000 → 0000100010
 001 → 1011110000
 010 →
 011 → 11101
 100 →
 101 → 00100

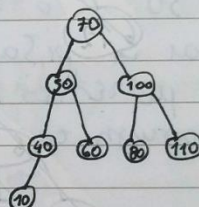
Διαγραφή του 11101:

000 → 0000100010
 001 →
 010 → 1011110000
 011 →
 100 → 00100
 101 →

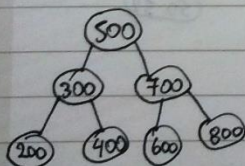
γ) Διαγραφή του 120:



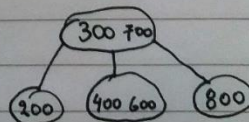
Είσαγωγή του 10:



δ) Το δέντρο με 7 2-κόμβους:

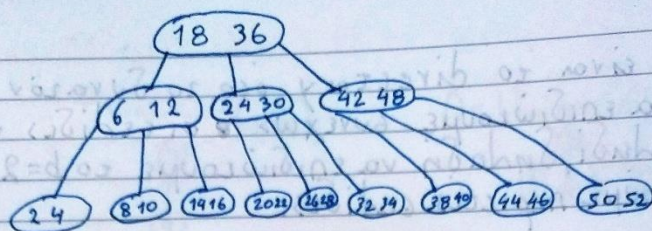


Μετά τη διαγραφή του 500: →



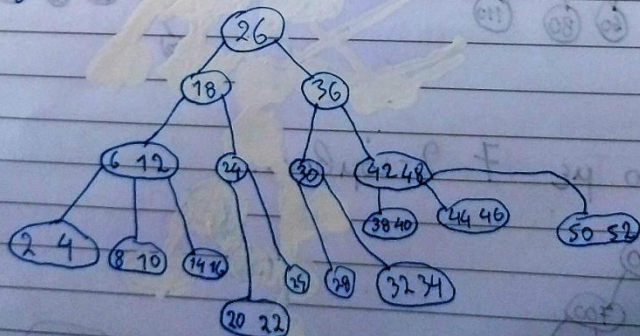
Διαγράφεται η ρίζα και χωνεύονται τους 300 700 βέε ένα 3-κόμβος ώστε να γίνει η ρίζα αλλά παρατηρούμε ότι έχει 4 παιδιά με την ένωση του 400 600 βέε ένα 3-κόμβος λύνεται το πρόβλημα και το δέντρο είναι 2-3 με την κατάλληλη ταξινόμηση

ε.)

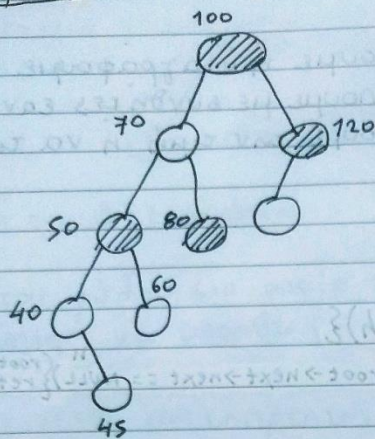


Εισαγωγή του 25 αναλυτικά:

Εύρεση του φύλου για οποίο θα επηρεάσει με την ιδιότητα της ταξινόμησης να εισαχθεί ο νέος κόμβος. Αυτός είναι ο (26, 28) ο οποίος επειδή είναι 3-κόμβος, δίδεται το βγόμενο, γίνεται μη επηρεαστός και τότε τον βγάμε σε 2 κόμβους ένα με το 25, ένα με το 28 και έναν πατρικό του (24, 30) ο οποίος όμως δεν μπορεί να δεχθεί τον 26 γιατί είναι παλι 3-κόμβος άρα θα βγάει σε δύο έναν με το 24 έναν με τον 30 και ο 26 θα πάει στον πατρικό του (18, 36) ο οποίος είναι και ρίζα 3-κόμβος θα βγάει σε δύο, τον 18 και τον 36 με τελικό πατέρα και ρίζα το 26. Τελική μορφή μετά την εισαγωγή του 25.

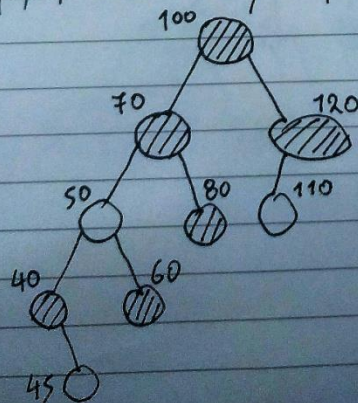


Άσκηση 25



Θεωρούμε τον 45 αρχικά κόκκινο και είμαστε στην πρώτη περίπτωση αποκατάστασης χρωματισμού. Ο πατέρας του 45 είναι αριστερό παιδί του παππού του. Ο 45 και ο πατέρας του είναι κόκκινοι, άρα χρειαζόμαστε αποκατάσταση. Ο αδελφικός κόμβος 60 του πατέρα 40 του 45 είναι κόκκινος, άρα κάνουμε τον παππού του 45 κόκκινο και τα παιδιά του μαύρα. Όμως έτσι ο 70 που είναι κόκκινος δεν θα έχει 2 μαύρα παιδιά, άρα ο 70 γίνεται τελικά μαύρος και έχουμε ένα κόκκινο-μαύρο δέντρο.

Τελική μορφή του δέντρου μετά την εμβάθυνση:



Άσκηση 3

- α) Κάθε φορά που είδα γούμε ή διαγράφουμε ένα κόμβο μπορούμε να βλέπουμε με συνθήκες εάν αλλάξει το μαύρο ύψος και να αυξάνουμε την τιμή ή να την μειώσουμε ανάλογα.

γ) void PreOrder(root, bh, node, bhh) {
 if (node == NULL) return;
 if (node->color == 'BLACK') { bhh--; node->bh = bhh; }
 PreOrder(root, bh, node->lc, bhh);
 PreOrder(root, bh, node->rc, bhh);
}

