Δομές Δεδομένων

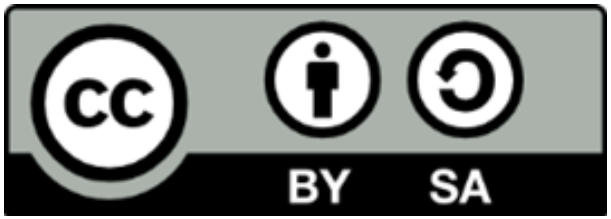
Ενότητα 9: Τα ΔΔΑ ως Αναδρομικές Δομές Δεδομένων-Εφαρμογή Δυαδικών Δέντρων: Κωδικοί Huffman

Καθηγήτρια Μαρία Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοποί ενότητας

- Να γνωρίζουν τον αναδρομικό ορισμό του ΔΔΑ
- Να υλοποιούν τις αναδρομικές διαδικασίες αναζήτησης, εισαγωγής, διαγραφής, διάσχισης (LNR, NLR. LRN) σε ΔΔΑ
- Να γνωρίζουν για τους κώδικες μεταβλητού μήκους
- Τα άμεσα αποκωδικοποιήσιμα συστήματα
- Το δένδρο Huffman και τον αντίστοιχο αλγόριθμο για την κωδικοποίηση και αποκωδικοποίηση

Περιεχόμενα ενότητας[1]

- ΔΔΑ ως αναδρομική δομή δεδομένων
- Αναδρομικός ορισμός δυαδικού δέντρου
- Παράδειγμα αναδρομικής διάσχισης
- Αναδρομική διάσχιση δέντρου
- Διαδικασία ενδοδιατεταγμένης διάσχισης δέντρου
- Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου
- Διαδικασία αναδρομικής αναζήτησης σε ΔΔΑ
- Αναδρομική διαδικασία εισαγωγής στοιχείου
- Αναδρομική διαδικασία διαγραφής στοιχείου

Περιεχόμενα ενότητας[2]

- Κώδικας μεταβλητού μήκους
- Κώδικας μεταβλητού μήκους: αναμενόμενο μήκος
- Άμεσα αποκωδικοποιήσιμα συστήματα
- Αλγόριθμος Huffman
- Εφαρμογή του αλγορίθμου Huffman
- Αλγόριθμος αποκωδικοποίησης Huffman
- Εφαρμογή αλγόριθμου αποκωδικοποίησης Huffman

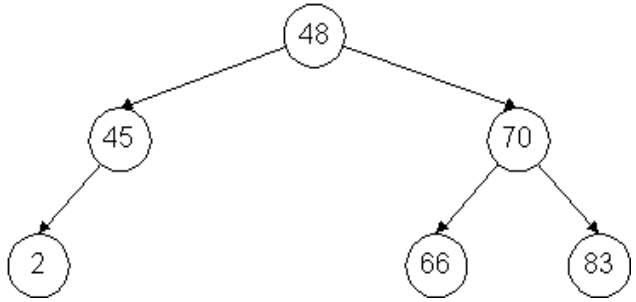
Τα ΔΔΑ ως Αναδρομικές Δομές Δεδομένων

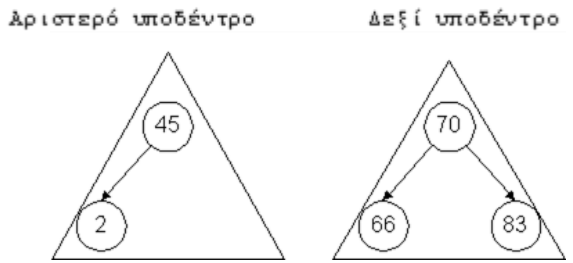
ΔΔΑ ως αναδρομική δομή δεδομένων

- Ένα δυαδικό δέντρο μπορεί πολύ φυσικά να οριστεί ως μια αναδρομική **δομή δεδομένων (recursive data structure)**.

ΔΔΑ ως αναδρομική δομή δεδομένων

- Στη ρίζα του δέντρου βρίσκεται ο αριθμός 48 και υπάρχουν δείκτες προς τους κόμβους με αριθμούς 45 και 70, καθένας από τους οποίους είναι ρίζα ενός δυαδικού υποδέντρου, όπως δείχνουν και τα ακόλουθα σχήματα:
- Ας πάρουμε τώρα το αριστερό υποδέντρο, που έχει ρίζα τον κόμβο με τον αριθμό 45 και ένα αριστερό παιδί, αλλά δεν έχει δεξί.
- Μπορούμε να θεωρήσουμε ότι αυτός ο κόμβος έχει δείκτες προς δυο δυαδικά υποδέντρα, ένα δεξί υποδέντρο και ένα αριστερό υποδέντρο (κενό), με την προϋπόθεση ότι επιτρέπονται τα κενά δέντρα:





Αναδρομικός ορισμός δυαδικού δέντρου -1-

- Επειδή το αριστερό υποδέντρο αποτελείται μόνο από έναν κόμβο, τον κόμβο με τον αριθμό 2, θα έχει κενό δεξί υποδέντρο και κενό αριστερό υποδέντρο.
- Έτσι, λοιπόν, ένα δυαδικό δέντρο μπορεί να οριστεί αναδρομικά ως εξής:
- Ένα δυαδικό δέντρο είτε

  α. είναι κενό

  είτε

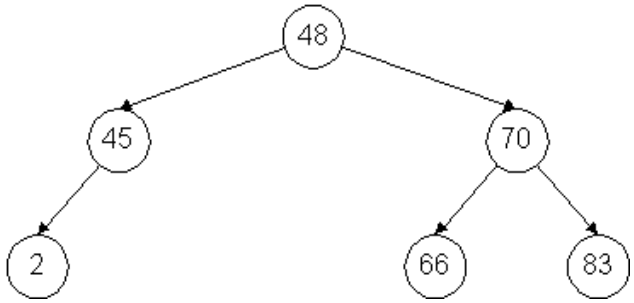
  β. αποτελείται από έναν κόμβο, που ονομάζεται ρίζα, ο οποίος έχει δείκτες προς δύο δυαδικά υποδέντρα, που ονομάζονται αριστερό υποδέντρο και δεξί υποδέντρο.

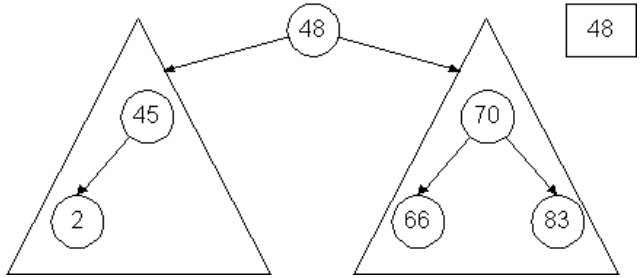
Αναδρομικός ορισμός δυαδικού δέντρου -2-

- Εξαιτίας της αναδρομικής φύσης των δυαδικών δέντρων, πολλές από τις βασικές λειτουργίες τους μπορούν να εκτελεστούν πολύ απλά χρησιμοποιώντας αναδρομικούς αλγόριθμους.
- Έστω, για παράδειγμα, η λειτουργία της διάσχισης, κατά την οποία μετακινούμαστε μέσα στο δυαδικό δέντρο και επισκεπτόμαστε κάθε κόμβο ακριβώς μια φορά.
- Υποθέτουμε ότι η σειρά με την οποία επισκεπτόμαστε κάθε κόμβο δεν έχει σημασία, αλλά είναι σημαντικό να επισκεφτούμε όλους τους κόμβους και να επεξεργαστούμε τις πληροφορίες που περιέχονται σ' αυτούς ακριβώς μια φορά.
- Ένα απλό αναδρομικό σχήμα είναι να διασχίσουμε το δυαδικό δέντρο ως εξής:
  - Επίσκεψη της ρίζας και επεξεργασία των περιεχομένων της.
  - Διάσχιση του αριστερού υποδέντρου.
  - Διάσχιση του δεξιού υποδέντρου.

Παράδειγμα αναδρομικής διάσχισης -1-

- εμφανίζοντας τα περιεχόμενα κάθε κόμβου που επισκεπτόμαστε,
- ξεκινάμε από τη ρίζα και εμφανίζουμε τον αριθμό 48, όπως δείχνει και το διπλανό σχήμα:
- Εν συνεχεία πρέπει να διασχίσουμε πρώτα το αριστερό υποδέντρο και έπειτα το δεξί.
- Όταν ολοκληρωθεί η διάσχιση και των δύο υποδέντρων θα έχουμε διασχίσει όλο το δυαδικό δέντρο.





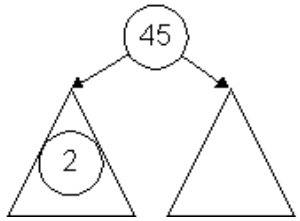
Παράδειγμα αναδρομικής διάσχισης -2-

- Τώρα το πρόβλημα έχει περιοριστεί στο να διασχίσουμε δύο μικρότερα δυαδικά δέντρα.
- Παίρνουμε πρώτα το αριστερό υποδέντρο (Σχήμα Β) και επισκεπτόμαστε τη ρίζα του, οπότε εμφανίζεται ο αριθμός 45.

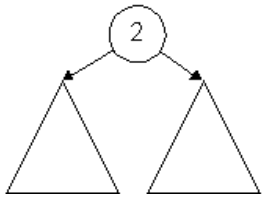
  Στη συνέχεια θα διασχίσουμε πρώτα το αριστερό υποδέντρο του και έπειτα το δεξί.
- Για να διασχίσουμε το αριστερό υποδέντρο, επισκεπτόμαστε τη ρίζα του και εμφανίζουμε τον αριθμό 2.

  Τώρα πρέπει να διασχίσουμε πρώτα το αριστερό και στη συνέχεια το δεξί υποδέντρο.

  Όπως φαίνεται και από το σχήμα, και τα δύο υποδέντρα είναι κενά οπότε δεν χρειάζεται να κάνουμε τίποτα, δηλαδή η διάσχιση του δέντρου του Σχήματος Γ έχει ολοκληρωθεί.



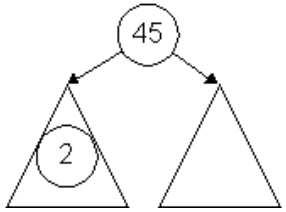
48 | 45



48 | 45 | 2

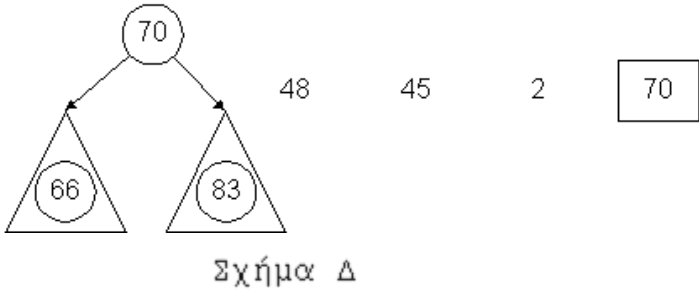
Παράδειγμα αναδρομικής διάσχισης -3-

- Τώρα πρέπει να διασχίσουμε το δεξί υποδέντρο του δέντρου του Σχήματος Β. Το υποδέντρο αυτό είναι κενό, οπότε και πάλι δεν κάνουμε τίποτα. Επομένως, η διάσχιση του αριστερού υποδέντρου της ρίζας του αρχικού δέντρου έχει ολοκληρωθεί και



- μένει να διασχίσουμε το δεξί υποδέντρο. Ξεκινάμε πάλι από τη ρίζα αυτού του υποδέντρου και εμφανίζουμε τον αριθμό 70, που είναι αποθηκευμένος εκεί, όπως φαίνεται και στο Σχήμα Δ.

  Στη συνέχεια θα διασχίσουμε το αριστερό και το δεξί υποδέντρο.



Σχήμα Δ

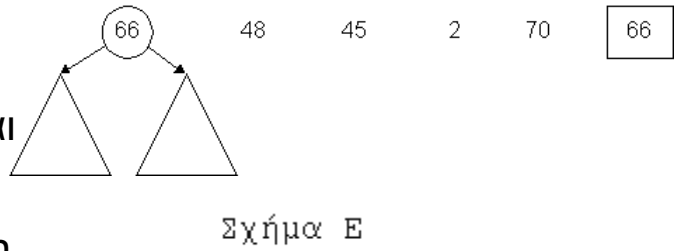
Παράδειγμα αναδρομικής διάσχισης -4-

- Πρώτα θα διασχίσουμε το αριστερό υποδέντρο του παραπάνω σχήματος, οπότε εμφανίζουμε τον αριθμό 66, που βρίσκεται στη ρίζα του.
- Ακολουθεί η διάσχιση του αριστερού υποδέντρου και του δεξιού υποδέντρου. Όπως φαίνεται και από το Σχήμα Ε, τα υποδέντρα αυτά είναι κενά, πράγμα που σημαίνει ότι δεν έχουμε να κάνουμε τίποτα και η διάσχιση του δέντρου του Σχήματος Ε έχει ολοκληρωθεί.
- Τώρα μας μένει να διασχίσουμε το δεξί υποδέντρο του Σχήματος Δ.

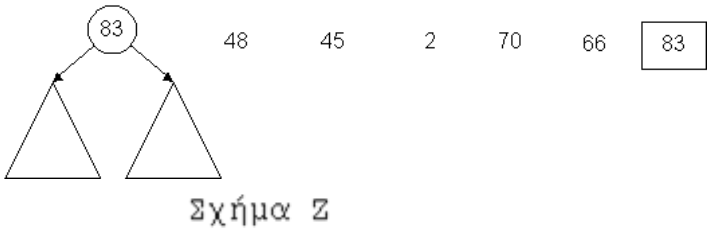
  Εμφανίζουμε τον αριθμό 83 της ρίζας του και στη συνέχεια πρέπει να διασχίσουμε το αριστερό και το δεξί υποδέντρο, όπως φαίνονται και στο Σχήμα Ζ.

  Επειδή και τα δύο αυτά δέντρα είναι κενά, δεν κάνουμε τίποτα.

  Η διάσχιση του δέντρου του Σχήματος Ζ έχει ολοκληρωθεί.



Σχήμα Ε



Σχήμα Ζ

Αναδρομική διάσχιση δέντρου

- Όπως φαίνεται, από το προηγούμενο παράδειγμα, η διάσχιση ενός δυαδικού δέντρου αναδρομικά απαιτεί τρία βασικά βήματα, τα οποία συμβολίζουμε ως N, L και R:

  N (Node): Επίσκεψη ενός κόμβου.

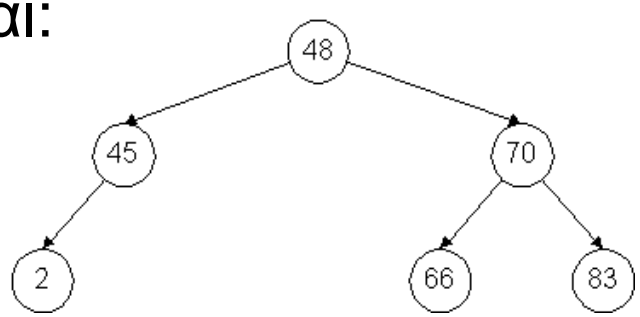
  L (Left): Διάσχιση του αριστερού υποδέντρου ενός κόμβου.

  R (Right): Διάσχιση του δεξιού υποδέντρου ενός κόμβου.

- Παρόλο που στο παράδειγμά μας εκτελέσαμε τα βήματα με αυτή τη σειρά, στην πραγματικότητα έχουμε έξι διατάξεις όσον αφορά τη σειρά εκτέλεσης των βημάτων:
- **LNR, NLR, LRN, , NRL, RNL , RLN**

Αναδρομική διάσχιση δέντρου: διάταξη LNR

Αν, για παράδειγμα εφαρμοστεί η διάταξη LNR, τότε ο αντίστοιχος αλγόριθμος διάταξης είναι:



Αν το δυαδικό δέντρο είναι κενό τότε
  Μην κάνεις τίποτα

Αλλιώς

  L:Διάσχιση του αριστερού υποδέντρου

  N:Επίσκεψη της ρίζας

  R:Διάσχιση του δεξιού υποδέντρου

Τέλος_αν

Αν εφαρμόσουμε τον παραπάνω αλγόριθμο στο δυαδικό δέντρο που χρησιμοποιήσαμε και προηγουμένως, η διάσχισή του εμφανίζει τους κόμβους με τη σειρά 2, 45, 38, 76, 70, 23.

Αναδρομική διάσχιση δέντρου: διατάξεις

Από τις προαναφερθείσες διατάξεις οι πιο σημαντικές είναι οι τρεις πρώτες, όπου πρώτα διασχίζεται το αριστερό υποδέντρο και μετά το δεξί, και έχουν τις εξής συγκεκριμένες ονομασίες:

LNR: ενδοδιατεταγμένη (inorder)

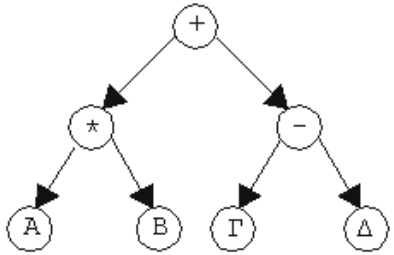
NLR: προδιατεταγμένη (preorder)

LRN: μεταδιατεταγμένη (postorder)

Δέντρα παράστασης -1-

Μια αριθμητική έκφραση μπορεί να παρασταθεί με ένα δυαδικό δέντρο, στο οποίο τα φύλλα περιέχουν τελεστέους και οι εσωτερικοί κόμβοι τελεστές. Ένα τέτοιο δέντρο ονομάζεται δέντρο παράστασης (expression tree).

Δέντρα παράστασης -2-



Παράδειγμα: η αριθμητική παράσταση

Α * Β + Γ - Δ

μπορεί να παρασταθεί με το διπλανό δυαδικό δέντρο:

όπου κάθε τελεστέος παριστάνεται σαν ένα παιδί ενός κόμβου πατέρα, που παριστάνει τον αντίστοιχο τελεστή.

Με μια ενδοδιατεταγμένη διάσχιση του παραπάνω δέντρου παράστασης προκύπτει η ενδοθεματική έκφραση: Α * Β + Γ - Δ

Μια προδιατεταγμένη διάσχιση του ίδιου δέντρου οδηγεί στην προθεματική έκφραση: + * Α Β - Γ Δ

Τέλος, μια μεταδιατεταγμένη διάσχιση του δέντρου μας δίνει την μεταθεματική έκφραση: Α Β * Γ Δ - +

Διαδικασία ενδοδιατεταγμένης διάσχισης δέντρου -1-

```
void RecBSTInorder(BinTreePointer Root)
/*Δέχεται:  Ένα δυαδικό δέντρο με το δείκτη
   Root να δείχνει στην ρίζα του.
  Λειτουργία: Εκτελεί ενδοδιατεταγμένη
   διάσχιση του δυαδικού δέντρου και
   επεξεργάζεται κάθε κόμβο ακριβώς μια
   φορά.
  Επιστρέφει: Εξαρτάται από το είδος της
   επεξεργασίας.*/
```

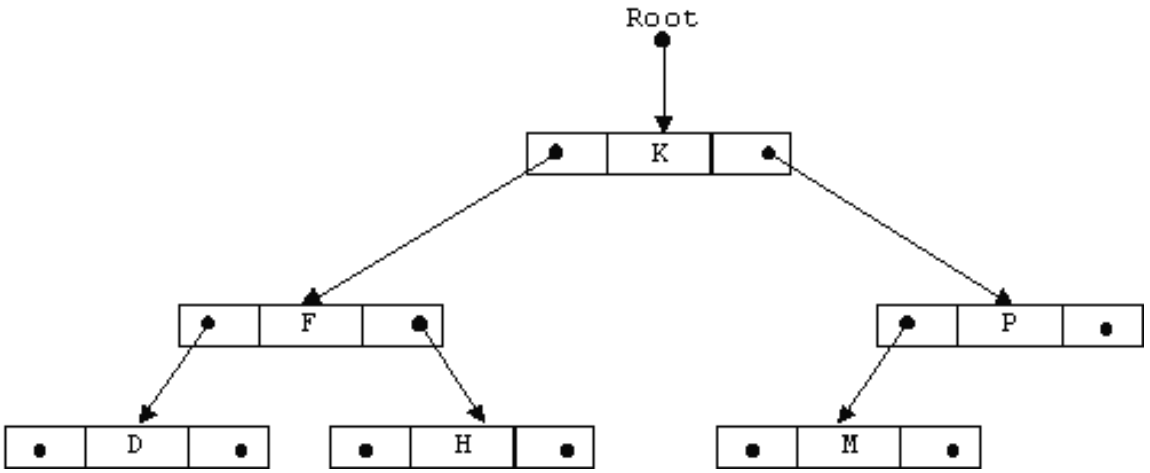
Διαδικασία ενδοδιατεταγμένης διάσχισης δέντρου -2-

```
{
    if (Root != NULL) {
        printf("L");
        RecBSTInorder(Root->LChild);
        printf("/%d ",Root->Data,"/");          //εδώ προστίθενται εντολές ανάλογα με
                                                    την επεξεργασία του κόμβου
        printf("R");
        RecBSTInorder(Root->RChild);
    }
    printf("U");
}
```

Για τα άλλα ήδη διατάξεων μπορούν να κατασκευαστούν διαδικασίες απλά αλλάζοντας τη σειρά των λειτουργιών L, N και R.

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου

Για να δούμε στην πράξη πώς λειτουργεί η παραπάνω διαδικασία, έστω ότι έχουμε το παρακάτω δυαδικό δέντρο:



Ο πίνακας που ακολουθεί δείχνει αναλυτικά την ενέργεια της διαδικασίας RecBSTInorderγια το παραπάνω δυαδικό δέντρο:

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου -1-

K	Κλήση της RecBSTInorder με δείκτη στη ρίζα (F) του αριστερού υποδέντρου.	
F	Κλήση της RecBSTInorder με δείκτη στη ρίζα (D) του αριστερού υποδέντρου.	
D	Κλήση της RecBSTInorder με δείκτη (NULL) στη ρίζα του αριστερού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
D	Εμφάνισε τα περιεχόμενα του κόμβου.	D
D	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του δεξιού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
D	Επιστροφή στον κόμβο πατέρα.	
F	Εμφάνισε τα περιεχόμενα του κόμβου.	F

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου -2-

F	Κλήση της RecBSTInorder με δείκτη στη ρίζα (H) του δεξιού υποδέντρου.	
H	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του αριστερού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
H	Εμφάνισε τα περιεχόμενα του κόμβου.	H
H	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του δεξιού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
H	Επιστροφή στον κόμβο πατέρα.	
F	Επιστροφή στον κόμβο πατέρα.	
K	Εμφάνισε τα περιεχόμενα του κόμβου.	K

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου -3-

Κ	Κλήση της RecBSTInorder με δείκτη στη ρίζα (Ρ) του δεξιού υποδέντρου.	
Ρ	Κλήση της RecBSTInorder με δείκτη στη ρίζα (Μ) του αριστερού υποδέντρου.	
Μ	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του αριστερού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
Μ	Εμφάνισε τα περιεχόμενα του κόμβου.	Μ
Μ	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του δεξιού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
Μ	Επιστροφή στον κόμβο πατέρα.	

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου -4-

P	Εμφάνισε τα περιεχόμενα του κόμβου.	P
P	Κλήση της RecBSTInorder με δείκτη στη ρίζα (NULL) του δεξιού υποδέντρου.	
τίποτα	Τίποτα, επιστροφή στον κόμβο πατέρα.	
P	Επιστροφή στον κόμβο πατέρα.	
K	Τέλος της διαδικασίας. Η διάσχιση ολοκληρώθηκε.	

Παράδειγμα ενδοδιατεταγμένης διάσχισης δέντρου -5-

Η ενδοθεματική διάσχιση που εκτελέσαμε για το παραπάνω δέντρο, εμφανίζει τα περιεχόμενα των κόμβων με αλφαβητική σειρά:

D, F, H, K, M, P

κι αυτό συμβαίνει γιατί, στην πραγματικότητα, πρόκειται όχι για ένα απλό δυαδικό δέντρο, αλλά για ένα ΔΔΑ.

Έτσι, λοιπόν, για κάθε κόμβο η τιμή που είναι αποθηκευμένη στο αριστερό παιδί του είναι μικρότερη από την τιμή του κόμβου αυτού και η τελευταία είναι μικρότερη από αυτήν που είναι αποθηκευμένη στο δεξί παιδί του κόμβου.

Επομένως, όλες οι τιμές που βρίσκονται σε κόμβους του αριστερού υποδέντρου ενός κόμβου είναι μικρότερες από την τιμή του κόμβου αυτού και η τιμή του κόμβου είναι με τη σειρά της μικρότερη από όλες τις τιμές των κόμβων του δεξιού υποδέντρου του.

Αφού, λοιπόν, μια ενδοθεματική διάσχιση είναι μια διάσχιση με διάταξη LNR, συμπεραίνουμε ότι η επίσκεψη των κόμβων γίνεται με αλφαβητική σειρά.

Διαδικασία αναδρομικής αναζήτησης σε ΔΔΑ -1-

```
void RecBSTSearch(BinTreePointer Root,
    BinTreeElementType KeyValue, boolean *Found,
    BinTreePointer *LocPtr)


/*Δέχεται: Ένα ΔΔΑ με το δείκτη Root να δείχνει στη ρίζα
    του και μια τιμή KeyValue.
  Λειτουργία: Εκτελεί αναδρομική αναζήτηση στο ΔΔΑ για
    έναν κόμβο με τιμή KeyValue στο πεδίο κλειδί του.
 Επιστρέφει: Η Found έχει τιμή TRUE και ο δείκτης
    LocPtr δείχνει στον κόμβο που περιέχει την τιμή
    KeyValue, αν η αναζήτηση είναι επιτυχής.
   Διαφορετικά η Found έχει τιμή FALSE.*/
```

Διαδικασία αναδρομικής αναζήτησης σε ΔΔΑ -2-

```
{
   if (BSTEmpty(Root))   *Found = FALSE;
   else
      if (KeyValue < Root->Data)
         RecBSTSearch(Root->LChild, KeyValue, &(*Found), &(*LocPtr));
      else
         if (KeyValue > Root->Data)
            RecBSTSearch(Root->RChild, KeyValue, &(*Found), &(*LocPtr));
         else
            {
               *Found = TRUE;  *LocPtr = Root;
            }
}
```

Αναδρομική διαδικασία εισαγωγής στοιχείου -1-

```
void RecBSTInsert(BinTreePointer *Root,
    BinTreeElementType Item)
/*Δέχεται: Ένα ΔΔΑ με το δείκτη Root να δείχνει
    στη ρίζα του και ένα στοιχείο Item.
  Λειτουργία:  Εισάγει αναδρομικά το στοιχείο
    Item στο ΔΔΑ.
  Επιστρέφει: Το τροποποιημένο ΔΔΑ με τον
    δείκτη Root να δείχνει στη ρίζα του.*/
```

Αναδρομική διαδικασία εισαγωγής στοιχείου -2-

```
{
   if (BSTEmpty(*Root)) {
      (*Root) = (BinTreePointer)malloc(sizeof (struct BinTreeNode));
      (*Root) ->Data = Item; (*Root) ->LChild = NULL;
      (*Root) ->RChild = NULL;
   }
   else if (Item < (*Root) ->Data)
                  RecBSTInsert(&(*Root) ->LChild,Item);
         else if (Item > (*Root) ->Data)
                  RecBSTInsert(&(*Root) ->RChild,Item);
               else
                  printf("To %d EINAI HDH STO DDA\n", Item);
}
```

Αναδρομική διαδικασία εισαγωγής διαγραφής -1-

void RecBSTDelete(BinTreePointer *Root, BinTreeElementType KeyValue)

/*Δέχεται: Ένα ΔΔΑ με το δείκτη Root να δείχνει στη ρίζα του και μια τιμή KeyValue.

Λειτουργία: Προσπαθεί αναδρομικά να βρει έναν κόμβο στο ΔΔΑ που να περιέχει την τιμή KeyValue στο πεδίο κλειδί του τμήματος δεδομένων του και, αν τον βρει, τον διαγράφει από το ΔΔΑ.

Επιστρέφει: Το τροποποιημένο ΔΔΑ με τον δείκτη Root να δείχνει στη ρίζα του/

Αναδρομική διαδικασία εισαγωγής διαγραφής -2-

```
{
    BinTreePointer TempPtr;
    if (BSTEmpty(*Root)) printf("to %d DeN BRE8HKe STO DDA\n",
    KeyValue);
    else
    if (KeyValue < (*Root)->Data)
                    RecBSTDelete(&((*Root)->LChild), KeyValue);
    else if (KeyValue > (*Root)->Data)
                    RecBSTDelete(&((*Root)->RChild), KeyValue);
          else if ((*Root)->LChild == NULL)  {
                TempPtr = *Root; *Root = (*Root)->RChild;
                 free(TempPtr );
              }
```

Αναδρομική διαδικασία εισαγωγής διαγραφής -3-

```
else if ((*Root)->RChild == NULL)  {
        TempPtr = *Root;  *Root = (*Root)-> LChild; free(TempPtr );
            }
    else {
            TempPtr = (*Root)->RChild;
            while (TempPtr->LChild != NULL)
                TempPtr = TempPtr->LChild;
            (*Root)->Data = TempPtr->Data;
            RecBSTDelete(&((*Root)->RChild), (*Root)->Data);
            }
}
```

Εφαρμογή Δυαδικών Δέντρων: Κωδικοί Huffman

Κώδικας μεταβλητού μήκους -1-

- Τα δυαδικά δέντρα μπορούν να χρησιμοποιηθούν σε πολλά προβλήματα κωδικοποίησης (encoding) και αποκωδικοποίησης (decoding).
- Ένα παράδειγμα είναι η κωδικοποίηση και αποκωδικοποίηση ενός μηνύματος που μεταφέρεται με σήματα Morse, όπου κάθε χαρακτήρας αναπαρίσταται με μια σειρά από τελείες και παύλες:
- Για παράδειγμα, το γράμμα Α συμβολίζεται ως .-, το γράμμα Β ως -..., το γράμμα C ως -.-., κλπ.
- Όταν χρησιμοποιείται ο κώδικας Morse, το πλήθος των συμβόλων που απαιτείται για την παράσταση κάθε γράμματος δεν είναι σταθερό.
- Το Α, για παράδειγμα θέλει δύο σύμβολα, ενώ το Β και το C θέλουν από τέσσερα.
- Ένας τέτοιος κώδικας ονομάζεται κώδικας μεταβλητού μήκους (variable-length code).

Κώδικας μεταβλητού μήκους -2-

- Στη συνέχεια θα ασχοληθούμε με τους Κώδικες Huffman, που είναι επίσης μεταβλητού μήκους κώδικες.
- Η βασική ιδέα στους κώδικες μεταβλητού μήκους είναι να χρησιμοποιούνται μικρότεροι κωδικοί για τους χαρακτήρες που εμφανίζονται πιο συχνά και μεγαλύτεροι γι' αυτούς που εμφανίζονται λιγότερο συχνά.
- Ο σκοπός είναι να μειωθεί το αναμενόμενο μήκος του κωδικού ενός χαρακτήρα ώστε να περιοριστεί και το πλήθος των bits που απαιτούνται για την μετάδοση του κωδικοποιημένου μηνύματος.

Κώδικας μεταβλητού μήκους: αναμενόμενο μήκος -1-

- Υποθέτουμε ότι για ένα σύνολο χαρακτήρων C_1, C_2,..., C_n, υπάρχει ένα σύνολο βαρών $w_1,w_2,\ldots,w_n$, που σχετίζεται με αυτούς τους χαρακτήρες, δηλαδή w_i είναι το βάρος που αντιστοιχεί στον χαρακτήρα C_i και είναι ένα μέτρο (πιθανότητα ή σχετική συχνότητα) του πόσο συχνά εμφανίζεται αυτός ο χαρακτήρας σε μηνύματα που πρόκειται να κωδικοποιηθούν.
- Αν με $l_1,l_2,\ldots,l_n$ συμβολίσουμε τα μήκη των κωδικών των χαρακτήρων $C_1,C_2,\ldots,C_n$ αντίστοιχα, τότε το αναμενόμενο μήκος του κώδικα για καθένα από τους χαρακτήρες αυτούς είναι:

  αναμενόμενο μήκος = $w_1l_1+w_2l_2+\ldots+w_nl_n = \sum_{i=1}^{n} w_i l_i$

Κώδικας μεταβλητού μήκους: αναμενόμενο μήκος -2-

Χαρακτήρας	Βάρος
A	0.2
B	0.15
C	0.05
D	0.15
E	0.45

- Έστω ότι έχουμε τους χαρακτήρες A, B, C, D και E και ότι οι πιθανότητες εμφάνισης αυτών των χαρακτήρων είναι τα βάρη που φαίνονται στον διπλανό πίνακα:
- Αναπαράσταση του συνόλου χαρακτήρων σε κώδικα Morse με τελείες και παύλες στην 2η στήλη, ενώ στην 3η στήλη οι τελείες έχουν αντικατασταθεί με 0 και οι παύλες με 1:

Χαρακτήρας	Κωδικός Morse(1)	Κωδικός Morse(2)
A	.-	01
B	-...	1000
C	-.-.	1010
D	-..	100
E	.	0

Αναμενόμενο μήκος κώδικα για τους παραπάνω πέντε χαρακτήρες:

$$2*0.2 + 4*0.15 + 4*0.05 + 3*0.15 + 1*0.45 = 2.1$$

Άμεσα αποκωδικοποιήσιμα συστήματα -1-

- Μια σημαντική ιδιότητα μερικών συστημάτων κωδικοποίησης είναι ότι είναι άμεσα αποκωδικοποιήσιμα (immediately decodable), δηλαδή καμία ακολουθία από bits, που αναπαριστά έναν χαρακτήρα, δεν αποτελεί πρόθεμα κάποιας μεγαλύτερης ακολουθίας bits, που να αναπαριστά έναν άλλο χαρακτήρα.
- Αυτό έχει ως συνέπεια, η λήψη μιας ακολουθίας από bits να αποκωδικοποιείται αμέσως στον αντίστοιχο χαρακτήρα, χωρίς να χρειάζεται να περιμένουμε επόμενα bits για να σχηματίσουμε μια μεγαλύτερη ακολουθία, που πιθανόν να αντιστοιχεί σε άλλον χαρακτήρα.
- Την ιδιότητα αυτή δεν την έχει ο κώδικας Morse κι αυτό φαίνεται στον πίνακα που παρουσιάσαμε.
- Ο κωδικός του D (100) είναι πρόθεμα του αντίστοιχου για τον χαρακτήρα B (1000) και ο κωδικός του χαρακτήρα E (0) αποτελεί πρόθεμα του χαρακτήρα A (01).

Άμεσα αποκωδικοποιήσιμα συστήματα -2-

- Μια κωδικοποίηση των χαρακτήρων A, B, C, D και E με το ίδιο μήκος κωδικών, όπως και παραπάνω, αλλά που είναι συγχρόνως άμεσα αποκωδικοποιήσιμη, φαίνεται στον πίνακα που ακολουθεί:
- Για την κωδικοποίηση χαρακτήρων με τρόπο ώστε να έχουν το μικρότερο αναμενόμενο μήκος και να αποκωδικοποιούνται αμέσως, μπορεί να χρησιμοποιηθεί ο αλγόριθμος του D. A. Huffman που παρουσιάζεται στη συνέχεια.

Χαρακτήρας	Κωδικός
A	01
B	0001
C	0000
D	001
E	1

Αλγόριθμος Huffman -1-

/*Δέχεται: Ένα σύνολο από n χαρακτήρες {C1,C2,...,Cn} και ένα σύνολο από βάρη {w1,w2,...,wn}, όπου wi είναι το βάρος του χαρακτήρα Ci.

Λειτουργία: Κατασκευάζει ένα δυαδικό κωδικό για το δοσμένο σύνολο χαρακτήρων, όπου το αναμενόμενο μήκος της ακολουθίας bits κάθε χαρακτήρα είναι το ελάχιστο.

Επιστρέφει: Μια συλλογή από n ακολουθίες bits που αναπαριστούν κωδικούς των χαρακτήρων.*/

Αλγόριθμος Huffman -2-

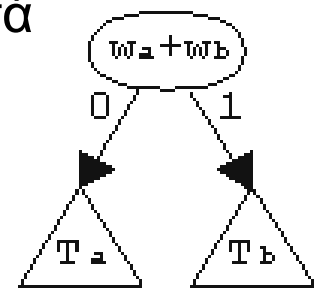
Αρχικοποίησε μια λίστα από δυαδικά δέντρα ενός κόμβου που να περιέχουν τα

βάρη w1,w2,...,wn, ένα για κάθε χαρακτήρα του συνόλου {C1,C2,...,Cn}

Για i από 1 μέχρι n-1

α. Βρες δύο δέντρα Τα και Τβ σ' αυτήν την λίστα με ρίζες τα ελάχιστα βάρη wα και wβ

β. Αντικατέστησε τα δύο αυτά δέντρα με ένα δυαδικό δέντρο του οποίου η ρίζα είναι wα+wβ, τα υποδέντρα του είναι τα Τα και Τβ και θέσε ετικέτες 0 και 1 στους δείκτες προς τα δύο αυτά υποδέντρα αντίστοιχα:



Τέλος_επανάληψης

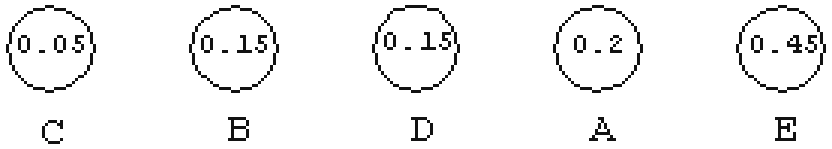
3. Ο κωδικός του χαρακτήρα Ci είναι η ακολουθία bits που σχηματίζεται από την διαδρομή στο τελικό δυαδικό δέντρο που ξεκινά από τη ρίζα και καταλήγει στο φύλλο Ci

Εφαρμογή του αλγορίθμου Huffman -1-

Ας εφαρμόσουμε τον αλγόριθμο Huffman στους χαρακτήρες

A, B, C, D και E με βάρη αυτά που δόθηκαν παραπάνω.

Ξεκινάμε κατασκευάζοντας μια λίστα από δυαδικά δέντρα ενός κόμβου, ένα για καθένα από τους πέντε χαρακτήρες:

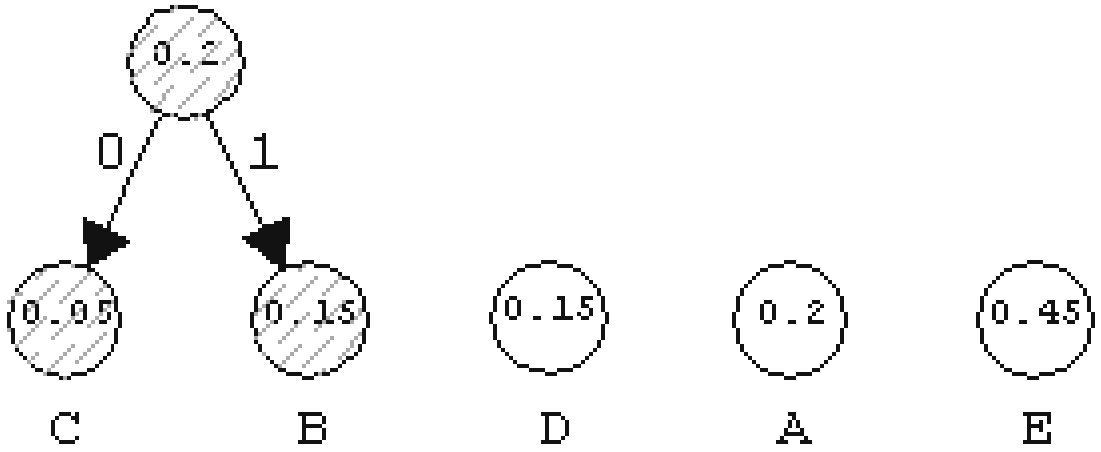


Επειδή στο βήμα 2.α του παραπάνω αλγόριθμου αναζητούνται τα δέντρα με τα ελάχιστα βάρη, διατάσσουμε τα δέντρα ενός κόμβου σε αύξουσα διάταξη.

Τα δύο μικρότερα βάρη είναι αυτά που αντιστοιχούν στους χαρακτήρες C και B (εναλλακτικά μπορούμε να πάρουμε τα βάρη των C και D, γιατί οι χαρακτήρες B και D έχουν ίδιο βάρος).

Εφαρμογή του αλγορίθμου Huffman -2-

Επομένως, τα δύο πρώτα δέντρα που επιλέγονται είναι τα δέντρα-κόμβοι που αντιστοιχούν στους χαρακτήρες C και B και κατασκευάζεται ένα δέντρο με βάρος 0.05+0.15=0.2 όπως δείχνει το ακόλουθο σχήμα:



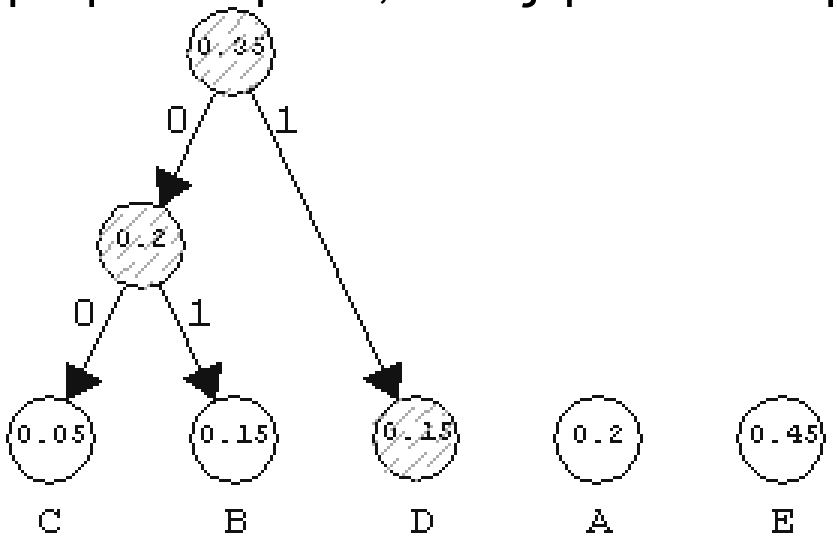
Εφαρμογή του αλγορίθμου Huffman -3-

Τώρα θα διαλέξουμε πάλι τα δύο μικρότερα βάρη από τη λίστα των τεσσάρων πλέον δυαδικών δέντρων του παραπάνω σχήματος.

Τα βάρη είναι τώρα 0.2, 0.15, 0.2 και 0.45.

Από αυτά διαλέγουμε το δέντρο με βάρος 0.15 (μικρότερο από όλα) και ένα από τα δύο δέντρα που έχουν βάρος 0.2, για να προκύψει ένα δέντρο με βάρος 0.15+0.2=0.35.

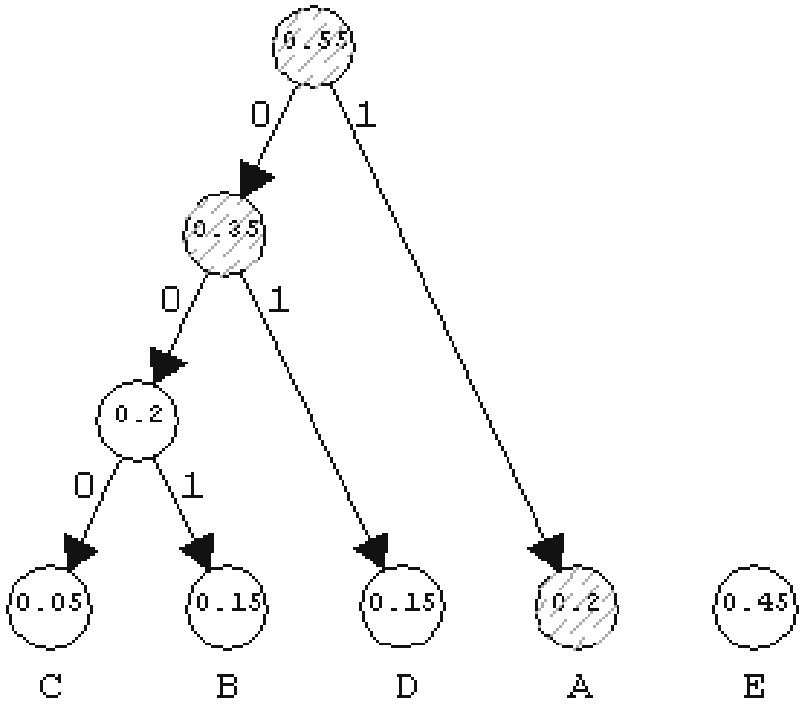
Έστω ότι διαλέγουμε το πρώτο, όπως φαίνεται παρακάτω



Εφαρμογή του αλγορίθμου Huffman -4-

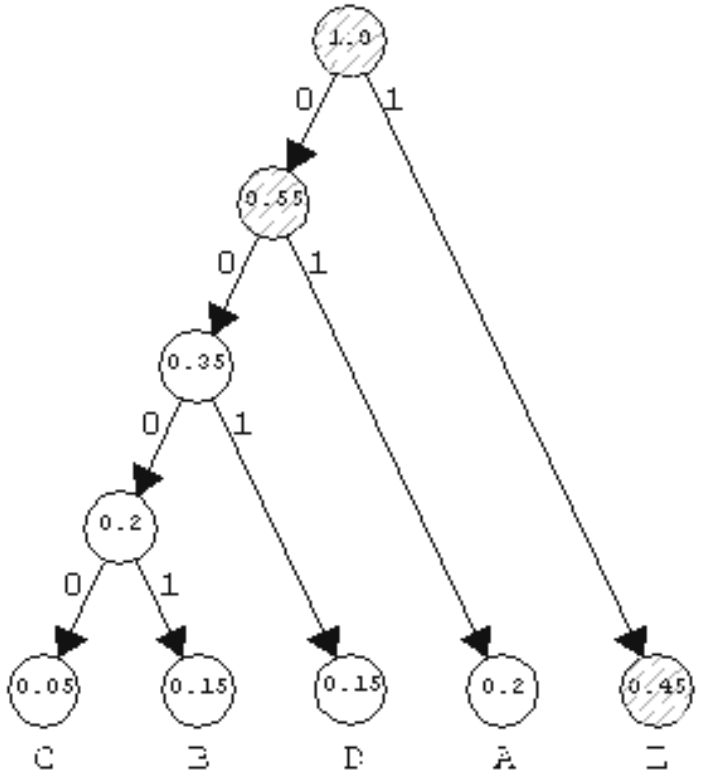
Από τα τρία δυαδικά δέντρα της λίστας επιλέγουμε πάλι τα δύο με τα μικρότερα βάρη.

Οι τιμές των βαρών είναι 0.35, 0.2 και 0.45, οπότε η μόνη μας επιλογή είναι να συνδυάσουμε τα δέντρα με βάρη 0.35 και 0.2 και να προκύψει ένα δέντρο με βάρος 0.35+0.2=0.55:



Εφαρμογή του αλγορίθμου Huffman -5-

Τώρα στη λίστα έχουν μείνει δύο δέντρα, τα οποία και συνδυάζονται για να προκύψει το τελικό δέντρο Huffman με βάρος 0.55+0.45=1.0 που φαίνεται παρακάτω:



Εφαρμογή του αλγορίθμου Huffman -6-

Οι κωδικοί Huffman που προκύπτουν για καθένα από τους πέντε χαρακτήρες είναι:

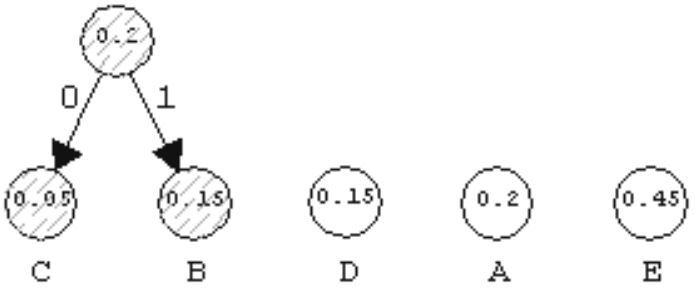
Χαρακτήρας	Κωδικός Huffman
A	01
B	0001
C	0000
D	001
E	1

Το αναμενόμενο μήκος κώδικα για τους χαρακτήρες αυτούς είναι πάλι:

$2*0.2 + 4*0.15 + 4*0.05 + 3*0.15 + 1*0.45 = 2.1$

Εφαρμογή του αλγορίθμου Huffman -7-

Το παραπάνω δέντρο Huffman δεν είναι το μοναδικό που θα μπορούσε να κατασκευαστεί για τους συγκεκριμένους χαρακτήρες και τα αντίστοιχα βάρη τους.

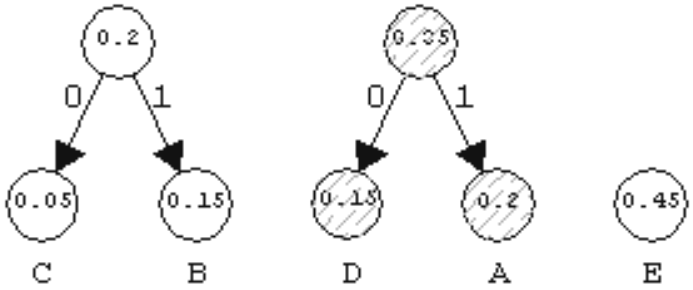


Μετά από την κατασκευή του δυαδικού δέντρου με βάρος 0.2:

θα μπορούσαμε να επιλέξουμε τα δέντρα που αντιστοιχούν στους χαρακτήρες D και A και να κατασκευάσουμε ένα δέντρο με βάρος 0.15+0.2=0.35, όπως φαίνεται στο παρακάτω σχήμα:

Το αναμενόμενο μήκος κώδικα για τους χαρακτήρες αυτούς είναι πάλι:

2*0.2 + 4*0.15 + 4*0.05 + 3*0.15 + 1*0.45 = 2.1



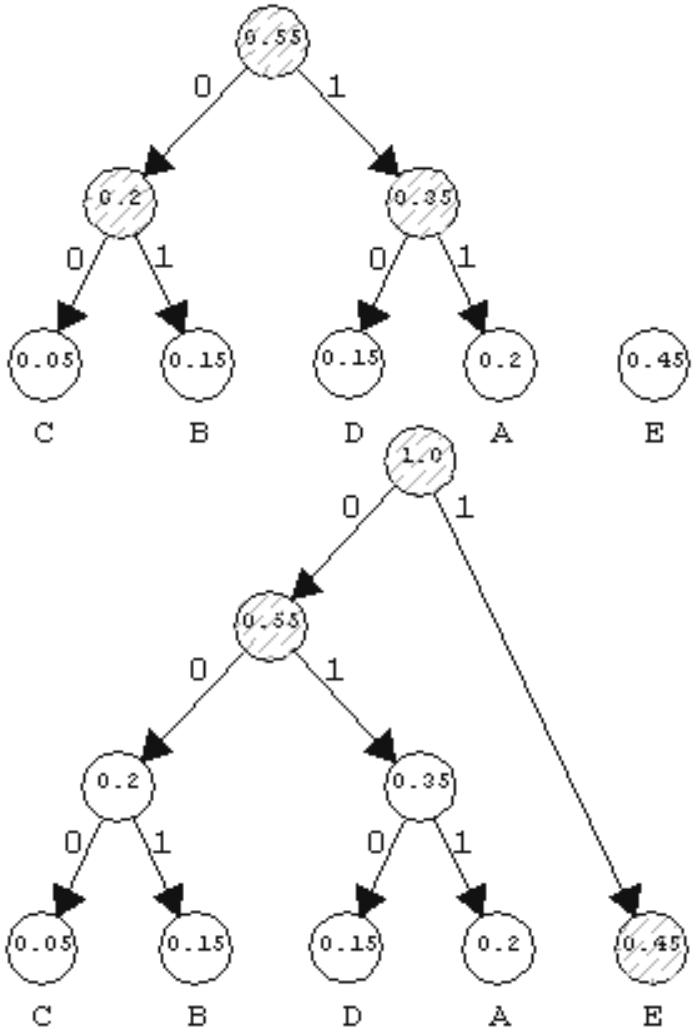
Εφαρμογή του αλγορίθμου Huffman -8-

Στη συνέχεια, από τη λίστα με τα τρία δέντρα, επιλέγουμε τα δύο με τα μικρότερα βάρη, οπότε προκύπτει το δέντρο που φαίνεται στο ακόλουθο σχήμα:

Το αναμενόμενο μήκος κώδικα για τους χαρακτήρες αυτούς είναι πάλι:

2*0.2 + 4*0.15 + 4*0.05 + 3*0.15 + 1*0.45 = 2.1

Τέλος, συνδυάζουμε τα δύο δέντρα με βάρη 0.55 και 0.45 και έχουμε κατασκευάσει το τελικό δέντρο Huffman, που διαφέρει από το προηγούμενο:



Εφαρμογή του αλγορίθμου Huffman -9-

Οι κωδικοί Huffman που προκύπτουν από το τελευταίο σχήμα είναι:

Χαρακτήρας	Κωδικός Huffman
A	011
B	001
C	000
D	010
E	1

Τέλος, συνδυάζουμε τα δύο δέντρα με βάρη 0.55 και 0.45 και έχουμε κατασκευάσει το τελικό δέντρο Huffman, που διαφέρει από το προηγούμενο:

Αλγόριθμος αποκωδικοποίησης Huffman -1-

/*Δέχεται: Ένα δέντρο Huffman και μια ακολουθία από bits που αναπαριστά ένα μήνυμα, το οποίο έχει κωδικοποηθεί με χρήση του δέντρου Huffman.

Λειτουργία: Αποκωδικοποιεί το μήνυμα.

Επιστρέφει: Το αποκωδικοποιημένο μήνυμα.*/

1. Αρχικοποίησε έναν δείκτη p να δείχνει στην ρίζα του δέντρου Huffman

Αλγόριθμος αποκωδικοποίησης Huffman -2-

Όσο δεν έχει βρεθεί το τέλος του μηνύματος επανάλαβε

α. Έστω ότι x είναι το επόμενο bit στην ακολουθία χαρακτήρων

β. Αν (x == 0) τότε p ← p->LChild

Αλλιώς p ← p->RChild

Τέλος_αν

γ. Αν ο p δείχνει σε φύλλο τότε

i. Εμφάνισε το χαρακτήρα που αντιστοιχεί σ' αυτό το φύλλο

ii. Να επαναφέρεις τον δείκτη p ώστε να δείχνει στην ρίζα του δέντρου Huffman

Τέλος_αν

Τέλος_επανάληψης

Εφαρμογή αλγόριθμου αποκωδικοποίησης Huffman -1-

Για να δούμε πώς εφαρμόζεται ο παραπάνω αλγόριθμος, έστω ότι θέλουμε να αποκωδικοποιήσουμε το μήνυμα:

0 0 1 1 0 1 0

που έχει κωδικοποιηθεί με χρήση του δεύτερου δέντρου Huffman που κατασκευάσαμε.

Ο δείκτης p ξεκινά από τη ρίζα και ακολουθεί το μονοπάτι 001 που οδηγεί στο γράμμα B:

0 0 1 1 0 1 0

B

Στη συνέχεια, ο δείκτης p επανέρχεται και δείχνει στη ρίζα του δέντρου. Το bit 1 που ακολουθεί οδηγεί κατευθείαν στο φύλλο που αντιστοιχεί στο χαρακτήρα E:

0 0 1 **1** 0 1 0

B E

Εφαρμογή αλγόριθμου αποκωδικοποίησης Huffman -2-

Ο δείκτης p ξεκινά πάλι από τη ρίζα του δέντρου και διαγράφει το μονοπάτι 010, που οδηγεί στο φύλλο για τον χαρακτήρα D:

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης