Κωδικοποίηση Huffman

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο

Huffman Coding

- Μέθοδος συμπίεσης δεδομένων: Μας δίνεται αρχείο με n διαφορετικούς χαρακτήρες. Θέλουμε να αντιστοιχίσουμε τον κάθε χαρακτήρα σε ένα δυαδικό κωδικό, έτσι ώστε να ελαχιστοποιηθεί ο συνολικός αριθμός των bits όλου του αρχείου.
- Ο κώδικας Huffman είναι ένας συγκεκριμένος τύπος βέλτιστου προθεματικού κώδικα που χρησιμοποιείται συνήθως για συμπίεση δεδομένων χωρίς απώλειες πληροφορίας.
- Ο David A. Huffman ανέπτυξε έναν (άπληστο) αλγόριθμο όταν ήταν μεταπτυχιακός φοιτητής στο MIT, και δημοσιεύτηκε στην εργασία του «A Method for the Construction of Minimum-Redundancy Codes» το 1952.

Huffman Coding

- Ο αλγόριθμος παράγει μία κωδικοποίηση βασισμένη στην πιθανότητα εμφάνισης του κάθε συμβόλου σε ένα κείμενο.
- Η κωδικοποίηση Huffman αποδεικνύεται βέλτιστη για ένα δεδομένο κείμενο.

Ορισμός του προβλήματος

- Είσοδος:

  Ένα σύνολο συμβόλων και τα βάρη τους (ανάλογα με τις πιθανότητες εμφανισης).

- Έξοδος:

  Ένας δυαδικός κώδικας χωρίς πρόθεμα (ένα σύνολο κωδικών λέξεων) με ελάχιστο αναμενόμενο μήκος κωδικών λέξεων (ισοδύναμα, ένα δέντρο με ελάχιστο σταθμισμένο μήκος διαδρομής από τη ρίζα).

Αποκωδικοποίηση μηνύματος με κώδικα μεταβλητού μήκους

- Σύμβολα {a,b,c,d}

- Μη διφορούμενοι κώδικες (prefix free – απροθεματικός)
- Κωδικοποίηση **A: 0**, **B: 10**, **C: 111**, **D: 110**
- Μήνυμα **00101111001100 AABCBADA**

- Διφορούμενοι κώδικες (non prefix free)
- Κωδικοποίηση **A: 0**, **B: 10**, **C: 11**, **D: 101**
- **B: 10** πρόθεμα του **D: 101**
- Μήνυμα **00101010001110 10 AABBBAACBB**
- Μήνυμα **00101010001110 10 AABDAAACDA**

Κωδικοποίηση Μεταβλητού Μήκους

- Έστω αρχείο που αποτελείται από τους χαρακτήρες α,β,γ,δ,ε,ζ,η και θ. Έστω 100 το πλήθος συμβόλων του κειμένου.
- Κωδικοποίση ASCII απαιτεί 100*7 =700 bits (για 8bit ASCII 800 bits)
- Βέλτιστος κώδικας σταθερού μήκους (προσαρμοσμένος στο πλήθος των συμβόλων) απαιτεί 3bits για κάθε σύμβολο ($2^3 = 8$). Επομένως 300 bits.
- Ταξινομούμε τα σύμβολα σε φθίνουσα σειρά συχνότητας:
- a: 40, b: 14, c: 13, d: 17, e: 10, f: 6.
- Ένας κώδικας μεταβλητού μήκους απαιτεί 236 bits a: 0, b: 101, c: 100, d: 111, e: 1101, f: 1100.
- Πλήθος bits 40*1+14*3+13*3+17*3+10*4+6*4=236 bits.

Κωδικοποίηση Μεταβλητού Μήκους

- Πιθανότητες εμφάνισης σε φθίνουσα σειρά:
- p_a: 0.40, p_d: 0.17, p_b: 0.14, p_c: 0.13, p_e: 0.10, p_f: 0.06.
- Μήκος ανά σύμβολο: $L_a=1$, $L_d=3$, $L_b=3$, $L_c=3$, $L_e=4$, $L_f=4$
- Μέσο μήκος ανά σύμβολο:

  $L=p_1*L_1 + p_2*L_2 + p_3*L_3 + p_4*L_4 + p_5*L_5 + p_6*L_6 = 2,36$

Αλγόριθμος

- Άπληστο κριτήριο: Σε κάθε βήμα εξάγουμε τα 2 σύμβολα με τη μικρότερη συχνότητα και τα συγχωνεύουμε ως παιδιά ενός νέου κόμβου.
- Ο νέος κόμβος έχει label το άθροισμα των συχνοτήτων.
- Εισάγουμε τον νέο κόμβο με την συχνότητά του στην ουρά προτεραιότητας.

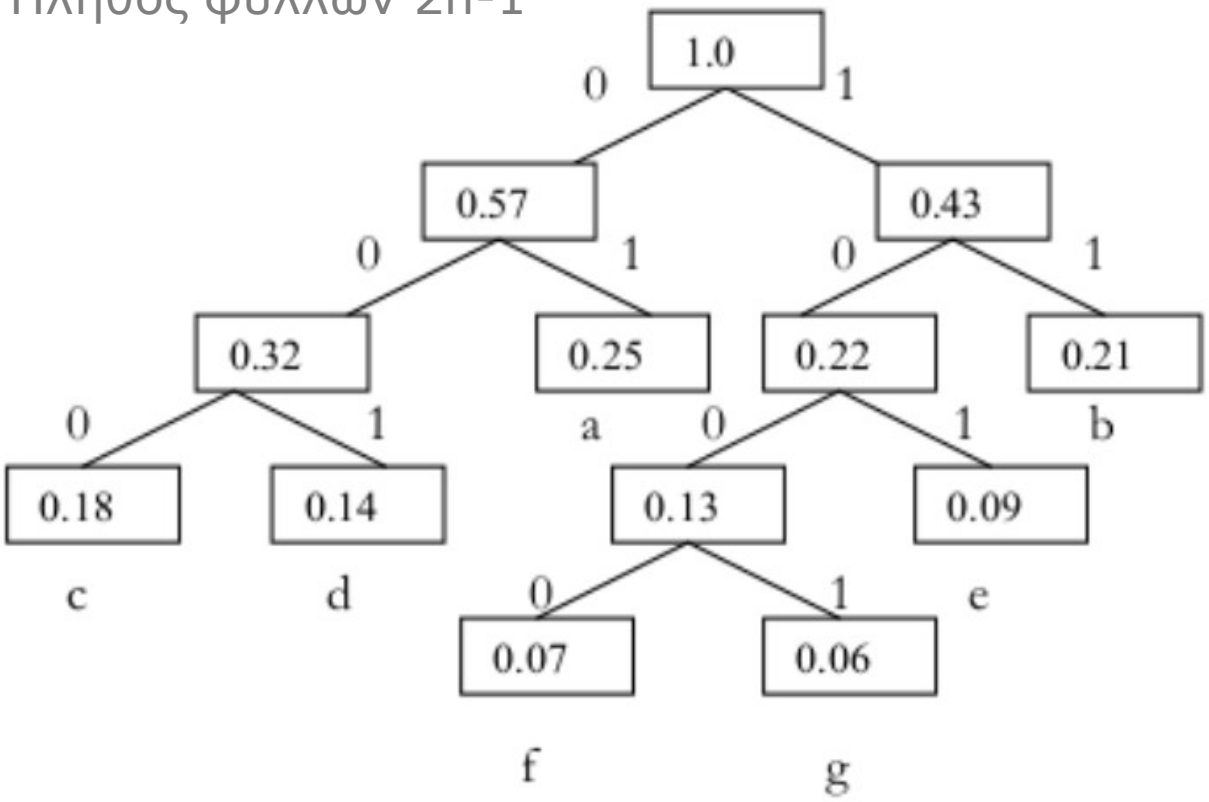
Παράδειγμα

Σύμβολα: *a, b, c, d, e, f, g*

Συχνότητες: 0.25, 0.21, 0.18, 0.14, 0.09, 0.07, 0.06

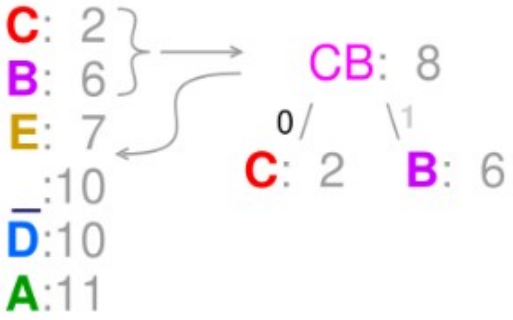
Κωδικοποίηση: a: 01, b: 11, c: 000, d: 001, e: 101, f: 1000, g: 1001

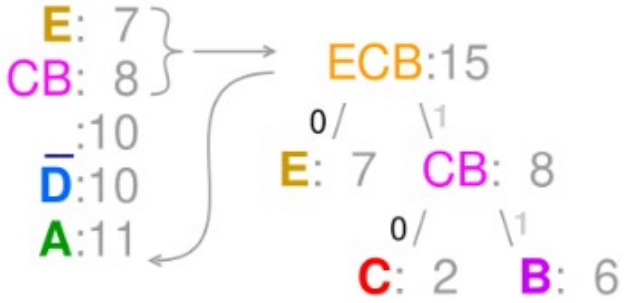
Πλήθος φύλλων 2n-1

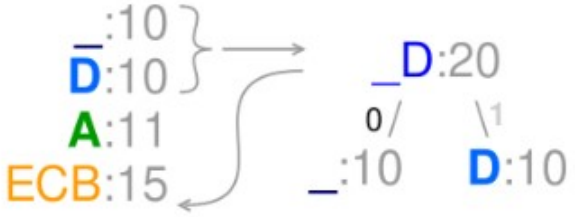


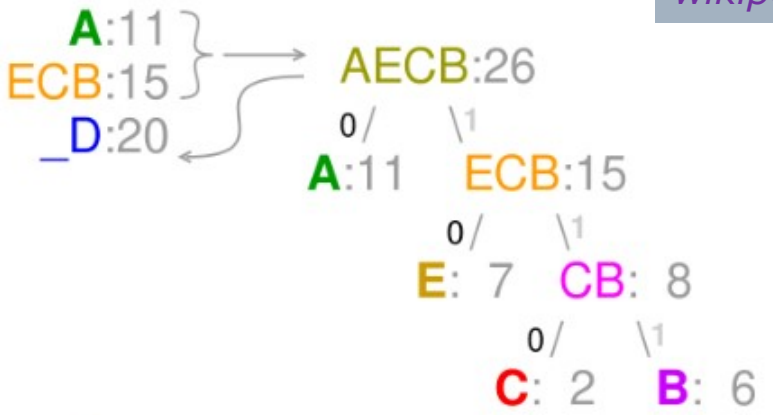
Παράδειγμα από wikipedia

1. "A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED"

2.


3.


4.


5.


6.


7.


8. "1000011101001000110010011101100111001001000111110010011111101111100010001111111010011100100101111110111010001111111001"

Αλγόριθμος

Αλγόριθμος 5.4 Κατασκευή του κώδικα Huffman

1: **Input**: character frequencies in nonincreasing order
2: **Output**: prefix code
3: **for** $i := 1$ to $n - 1$ **do**
4: merge last two subtrees
5: rearrange subtrees in nonincreasing order of root-probability
6: **end for**

O(nlogn)

Από το βιβλίο
«Τα θεμέλια της πληροφορικής»
Ζάχος Παγουρτζής

Διασπορά ως προς τη μέση τιμή

Σύμβολα: *a, b, c, d, e*

Συχνότητες: 0.4, 0.2, 0.2, 0.1, 0.1

Κωδικοποίηση με ισοπαλίες ελάχιστης διασποράς

a: 2 bits, b: 2, c: 2, d: 3, e: 3

Κωδικοποίηση με ισοπαλίες μη ελάχιστης διασποράς

a: 1 bit, b: 2, c: 3, d: 4, e: 4

Μέση τιμή μήκους και στις 2 περιπτώσεις:

0,4*2+0,2*2+0,2*2+0,1*3+0,1*3 = 2,2

0,4*1+0,2*2+0,2*3+0,1*4+0,1*4 = 2,2

Διαφορά στη διασπορά ως προς τη μέση τιμή

Βελτιστότητα

Είναι το δέντρο βέλτιστο;

Βέλτιστο δέντρο: το δέντρο ελαχίστου βάρους. Βάρος είναι το άθροισμα των γινομένων συχνότητα*μήκος. Ή άθροισμα συχνοτήτων κάθε κόμβου του δέντρου εκτός της ρίζας.

Λήμμα 1: Το δέντρο ενός βέλτιστου prefix code πρέπει να είναι πλήρες. Κάθε βέλτιστο δέντρο πρέπει να είναι πλήρες. (Όλοι οι εσωτερικοί κόμβοι πρέπει να έχουν 2 παιδιά).

Απόδειξη: Έστω πως κάποιος εσωτερικός κόμβος έχει μόνο ένα παιδί, τότε θα μπορούσαμε να διαγράψουμε αυτόν τον κόμβο και να τον αντικαταστήσουμε με το παιδί του. Αυτό θα μείωνε το κόστος της κωδικοποίησης.

Λήμμα 2: Έστω δύο σύμβολα x και y με τις μικρότερες συχνότητες, τότε υπάρχει βέλτιστο δέντρο κωδικοποίησης στο οποίο οι δύο χαρακτήρες είναι αδέλφια στο κατώτατο επίπεδο.

Βελτιστότητα

Λήμμα 2: Έστω δύο σύμβολα x και y με τις μικρότερες συχνότητες, τότε υπάρχει βέλτιστο δέντρο κωδικοποίησης στο οποίο οι δύο χαρακτήρες είναι αδέλφια στο κατώτατο επίπεδο.

Απόδειξη: Η απόδειξη του λήμματος χρησιμοποιεί *το επιχείρημα της ανταλλαγής*. Έστω T βέλτιστο δέντρο και έστω b και c τα δύο αδέλφια στο κατώτατο επίπεδο του δέντρου, τα οποία υπάρχουν αφού το T είναι πλήρες. Θεωρούμε χωρίς βλάβη της γενικότητας ότι $f(b) \leq f(c)$ και $f(x) \leq f(y)$. Αφού x, y οι χαρακτήρες με τις μικρότερες συχνότητες τότε $f(x) \leq f(b)$ και $f(y) \leq f(c)$, αφού b και c βρίσκονται στο κατώτατο επίπεδο τότε $depth(b) \geq depth(x)$ και $depth(c) \geq depth(y)$. Αλλάζοντας x με b φτιάχνουμε ένα καινούργιο δέντρο T'. $B(T) \leq B(T')$

$B(T') = B(T) - f(x)depth(x) - f(b)depth(b) + f(x)depth(b) + f(b)depth(x) = B(T) + (f(x) - f(b))(depth(b) - depth(x)) \leq B(T)$.

Βελτιστότητα

Επομένως $B(T') = B(T)$ δηλαδή T' βέλτιστο δέντρο. Ομοίως αλλάζουμε y με c και παίρνουμε T'' το οποίο είναι επίσης βέλτιστο και έχει την μορφή που ζητάμε.

Βελτιστότητα

Λήμμα 3: Έστω ένα δέντρο Τ σε αλφάβητο S, (n+1) συμβόλων, και έστω Τ' το δέντρο, σε αλφάβητο S' n συμβόλων, που προκύπτει αν κόψω τα δύο κάτω φύλλα (μικρότερης συχνότητας) και βάλω στον πρόγονό τους το άθροισμα των συχνοτήτων (και θεωρήσω τον πρόγονο νέο σύμβολο). Τότε ισχύει ότι:

Αν Τ' είναι βέλτιστο για S' <=>Τ είναι βέλτιστο για S

Απόδειξη: Από το Τ αφαιρώ τους δύο κόμβους μικρότερης συχνότητας έστω τον κόμβο 1 με συχνότητα f_1 και τον κόμβο 2 με συχνότητα f_2. Αντικαθιστώ τον πατέρα τους με ένα σύμβολο και το άθροισμα των συχνοτήτων των δύο κόμβων. Το βάρος του δέντρου Τ δίνεται από $w(T)= f_1 + f_2 +w(T')$. Αν Τ βέλτιστο τότε και Τ' βέλτιστο και αντιστρόφως.