Αλγόριθμοι και Πολυπλοκότητα

Αρχοντία Γιαννοπούλου
Όλγα Φουρτουνέλλη

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Άπληστοι Αλγόριθμοι

Κωδικοποίηση Huffman

Κωδικοποίηση Huffman

Δεδομένα: Μας δίνεται μία αλφαριθμητική ακολουθία δεδομένων
Στόχος: Αναπαράσταση των δεδομένων από τα δυφία (bits) 0, 1 με το συντομότερο δυνατό μήκος.

Κωδικοποίηση Huffman

Δεδομένα: Μας δίνεται μία αλφαριθμητική ακολουθία δεδομένων
Στόχος: Αναπαράσταση των δεδομένων από τα δυφία (bits) 0, 1 με το συντομότερο δυνατό μήκος.

- Τεχνική συμπίεσης δεδομένων.

Κωδικοποίηση Huffman

Δεδομένα: Μας δίνεται μία αλφαριθμητική ακολουθία δεδομένων
Στόχος: Αναπαράσταση των δεδομένων από τα δυφία (bits) 0, 1 με το συντομότερο δυνατό μήκος.

- Τεχνική συμπίεσης δεδομένων.
- Αναπαράσταση αλφαριθμητικής ακολουθίας δεδομένων με βάσει των συχνοτήτων εμφάνισης των χαρακτήρων.

Κωδικοποίηση Huffman

Δεδομένα: Μας δίνεται μία αλφαριθμητική ακολουθία δεδομένων
Στόχος: Αναπαράσταση των δεδομένων από τα δυφία (bits) 0, 1 με το συντομότερο δυνατό μήκος.

- Τεχνική συμπίεσης δεδομένων.
- Αναπαράσταση αλφαριθμητικής ακολουθίας δεδομένων με βάσει των συχνοτήτων εμφάνισης των χαρακτήρων.
- Εξάγει βέλτιστο τρόπο αναπαράστασης.

Κωδικοποίηση Huffman

Δεδομένα: Μας δίνεται μία αλφαριθμητική ακολουθία δεδομένων
Στόχος: Αναπαράσταση των δεδομένων από τα δυφία (bits) 0, 1 με το συντομότερο δυνατό μήκος.

- Τεχνική συμπίεσης δεδομένων.
- Αναπαράσταση αλφαριθμητικής ακολουθίας δεδομένων με βάσει των συχνοτήτων εμφάνισης των χαρακτήρων.
- Εξάγει βέλτιστο τρόπο αναπαράστασης.
- Δεν πρόκειται για κρυπτογράφηση δεδομένων.

Παράδειγμα

Στέλνουμε ένα email με n διαφορετικούς χαρακτήρες όπου για κάθε έναν από αυτούς γνωρίζουμε το πλήθος των εμφανίσεων του.

Παράδειγμα

Στέλνουμε ένα email με n διαφορετικούς χαρακτήρες όπου για κάθε έναν από αυτούς γνωρίζουμε το πλήθος των εμφανίσεων του.

Δεδομένα: n και $f_1, f_2, \ldots, f_n$ το πλήθος των εμφανίσεων του κάθε χαρακτήρα.
Ζητούμενο: Να αναπαραστήσουμε το κείμενο του email στο δυαδικό σύστημα.

Παράδειγμα

Στέλνουμε ένα email με n διαφορετικούς χαρακτήρες όπου για κάθε έναν από αυτούς γνωρίζουμε το πλήθος των εμφανίσεων του.

Δεδομένα: n και $f_1, f_2, \ldots, f_n$ το πλήθος των εμφανίσεων του κάθε χαρακτήρα.
Ζητούμενο: Να αναπαραστήσουμε το κείμενο του email στο δυαδικό σύστημα.

- Κωδικοποίηση των χαρακτήρων από συμβολοσειρές δυφίων.

Παράδειγμα

Στέλνουμε ένα email με n διαφορετικούς χαρακτήρες όπου για κάθε έναν από αυτούς γνωρίζουμε το πλήθος των εμφανίσεων του.

Δεδομένα: n και $f_1, f_2, \ldots, f_n$ το πλήθος των εμφανίσεων του κάθε χαρακτήρα.
Ζητούμενο: Να αναπαραστήσουμε το κείμενο του email στο δυαδικό σύστημα.

- Κωδικοποίηση των χαρακτήρων από συμβολοσειρές δυφίων.
- Η αποκωδικοποίηση πρέπει να είναι εφικτή.

Παράδειγμα

Στέλνουμε ένα email με n διαφορετικούς χαρακτήρες όπου για κάθε έναν από αυτούς γνωρίζουμε το πλήθος των εμφανίσεων του.

Δεδομένα: n και $f_1, f_2, \ldots, f_n$ το πλήθος των εμφανίσεων του κάθε χαρακτήρα.
Ζητούμενο: Να αναπαραστήσουμε το κείμενο του email στο δυαδικό σύστημα.

- Κωδικοποίηση των χαρακτήρων από συμβολοσειρές δυφίων.
- Η αποκωδικοποίηση πρέπει να είναι εφικτή.
- Το πλήθος των δυφίων που θα χρησιμοποιηθούν για όλο το κείμενο να είναι το ελάχιστο δυνατό.

Σχήματα κωδικοποίησης

- Σταθερό μήκος για κάθε χαρακτήρα (Μπορούμε να κωδικοποιήσουμε 2^k χαρακτήρες χρησιμοποιώντας μόνο συμβολοσειρές μήκους k)
  Πλεονέκτημα: Εύκολη αποκωδικοποίηση
  Μειονέκτημα: Δεν επιτυγχάνουν κωδικοποίηση ελαχίστου μήκους

Σχήματα κωδικοποίησης

- Σταθερό μήκος για κάθε χαρακτήρα (Μπορούμε να κωδικοποιήσουμε 2^k χαρακτήρες χρησιμοποιώντας μόνο συμβολοσειρές μήκους k)
  Πλεονέκτημα: Εύκολη αποκωδικοποίηση
  Μειονέκτημα: Δεν επιτυγχάνουν κωδικοποίηση ελαχίστου μήκους

Αν κάποιοι χαρακτήρες γνωρίζουμε ότι χρησιμοποιούνται πολύ πιο συχνά μπορούμε να το εκμεταλλευτούμε και να τους κωδικοποιήσουμε με λιγότερα δυφία έστω κι αν άλλοι (πιο αραιά εμφανιζόμενοι χαρακτήρες χρειαστούν κωδικοποίηση μεγαλύτερου μήκους);

Σχήματα κωδικοποίησης

- Σταθερό μήκος για κάθε χαρακτήρα (Μπορούμε να κωδικοποιήσουμε 2^k χαρακτήρες χρησιμοποιώντας μόνο συμβολοσειρές μήκους k)
  Πλεονέκτημα: Εύκολη αποκωδικοποίηση
  Μειονέκτημα: Δεν επιτυγχάνουν κωδικοποίηση ελαχίστου μήκους

  Αν κάποιοι χαρακτήρες γνωρίζουμε ότι χρησιμοποιούνται πολύ πιο συχνά μπορούμε να το εκμεταλλευτούμε και να τους κωδικοποιήσουμε με λιγότερα δυφία έστω κι αν άλλοι (πιο αραιά εμφανιζόμενοι χαρακτήρες χρειαστούν κωδικοποίηση μεγαλύτερου μήκους);

- Μεταβλητό μήκος για κάθε χαρακτήρα
  Πλεονέκτημα: Επιτυγχάνουν κωδικοποίηση ελαχίστου μήκους
  Μειονέκτημα: Σύνθετη κωδικοποίηση και αποκωδικοποίηση

Παράδειγμα

Μέγεθος αρχείου: 100.000 χαρακτήρες

Χαρακτήρες	**a**	**b**	**c**	**d**	**e**	**f**
Συχνότητα (σε χιλιάδες)	45	13	12	16	9	5
Κωδικός σταθερού μήκους	000	001	010	011	100	101
Κωδικός μεταβλητού μήκους	0	101	100	111	1101	1100

Παράδειγμα

Μέγεθος αρχείου: 100.000 χαρακτήρες

Χαρακτήρες	**a**	**b**	**c**	**d**	**e**	**f**
Συχνότητα (σε χιλιάδες)	45	13	12	16	9	5
Κωδικός σταθερού μήκους	000	001	010	011	100	101
Κωδικός μεταβλητού μήκους	0	101	100	111	1101	1100

Μήκος λέξης κωδικού σταθερού μήκους: 300.000 δυφία

Μήκος λέξης κωδικού μεταβλητού μήκους: 224.000 δυφία

$$45.000 \cdot 1 + 13.000 \cdot 3 + 12.000 \cdot 3 + 16.000 \cdot 3 + 9.000 \cdot 4 + 5.000 \cdot 4 = 224.000$$

Πώς αποκωδικοποιούμε;

Έστω το αλφάβητο a, b, c και οι κωδικοί τους:

- $a \rightarrow 01$
- $b \rightarrow 010$
- $c \rightarrow 1$

Σε ποιά λέξη αντιστοιχεί η συμβολοσειρά 0101;

Πώς αποκωδικοποιούμε;

Έστω το αλφάβητο a, b, c και οι κωδικοί τους:

- $a \to 01$
- $b \to 010$
- $c \to 1$

Σε ποιά λέξη αντιστοιχεί η συμβολοσειρά 0101;

Θα μπορούσε να είναι

- $aa = 01 \sqcup 01$
- $bc = 010 \sqcup 1$

Πώς το αποφεύγουμε;

Πώς αποκωδικοποιούμε;

Έστω το αλφάβητο a, b, c και οι κωδικοί τους:

- $a \to 01$
- $b \to 010$
- $c \to 1$

Σε ποιά λέξη αντιστοιχεί η συμβολοσειρά 0101;

Θα μπορούσε να είναι

- $aa = 01 \sqcup 01$
- $bc = 010 \sqcup 1$

Πώς το αποφεύγουμε;

- Είτε ορίζουμε σύμβολο κενού που 'σπάει' τις συμβολοσειρές που αντιστοιχούν σε ξεχωριστά γράμματα
- Είτε φροντίζουμε κανένας κωδικοποιημένος χαρακτήρας να μην είναι πρόθεμα κάποιου άλλου

Απροθεματική κωδικοποίηση

Μία απροθεματική κωδικοποίηση ενός συνόλου χαρακτήρων S είναι μία συνάρτηση c που αντιστοιχεί κάθε χαρακτήρα $x \in S$ σε μία ακολουθία δυφίων $c(x)$ έτσι ώστε για κάθε $x, y \in S$, με $x \neq y$, η συμβολοσειρά $c(x)$ να μην αποτελεί πρόθεμα της $c(y)$.

Για παράδειγμα,

- $c(a) = 11$
- $c(b) = 01$
- $c(d) = 001$
- $c(e) = 10$
- $c(f) = 000$

Ποιά είναι η λέξη 110100010010001;

Απροθεματική κωδικοποίηση

Μία απροθεματική κωδικοποίηση ενός συνόλου χαρακτήρων S είναι μία συνάρτηση c που αντιστοιχεί κάθε χαρακτήρα $x \in S$ σε μία ακολουθία δυφίων $c(x)$ έτσι ώστε για κάθε $x, y \in S$, με $x \neq y$, η συμβολοσειρά $c(x)$ να μην αποτελεί πρόθεμα της $c(y)$.

Για παράδειγμα,

- $c(a) = 11$
- $c(b) = 01$
- $c(d) = 001$
- $c(e) = 10$
- $c(f) = 000$

Ποιά είναι η λέξη 110100010010001;

Μονοσήμαντα μπορούμε να δούμε ότι η παραπάνω συμβολοσειρά αντιστοιχεί στην λέξη *abfeed*.

Κόστος κωδικοποίησης

Το κόστος μίας κωδικοποίησης c είναι το άθροισμα

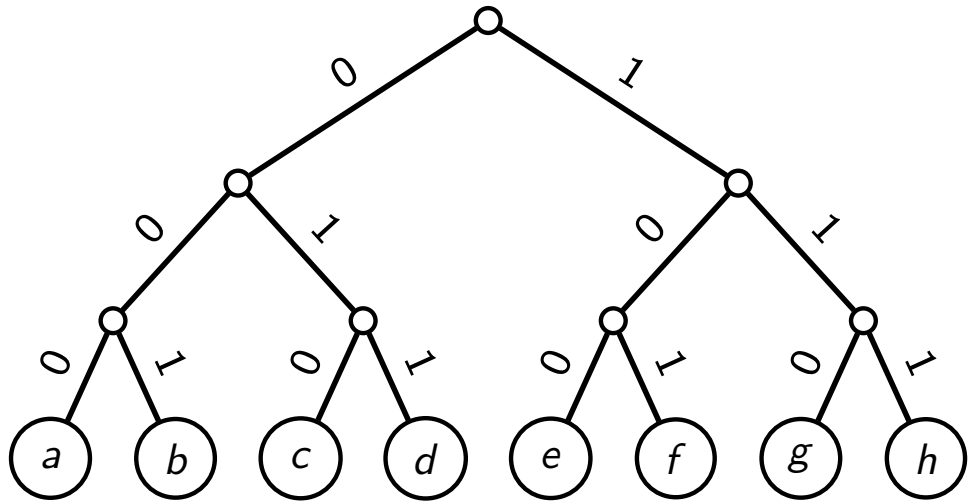
$$\mathbf{cost}(c) = \sum_{x \in S} f_x \cdot |c(x)|$$

Άρα το ζητούμενό μας είναι μία απροθεματική κωδικοποίηση που ελαχιστοποιεί την παραπάνω ποσότητα.

Αναπαράσταση κωδικοποιήσεων σε δυαδικά δέντρα

Κωδικοποίηση σταθερού μήκους

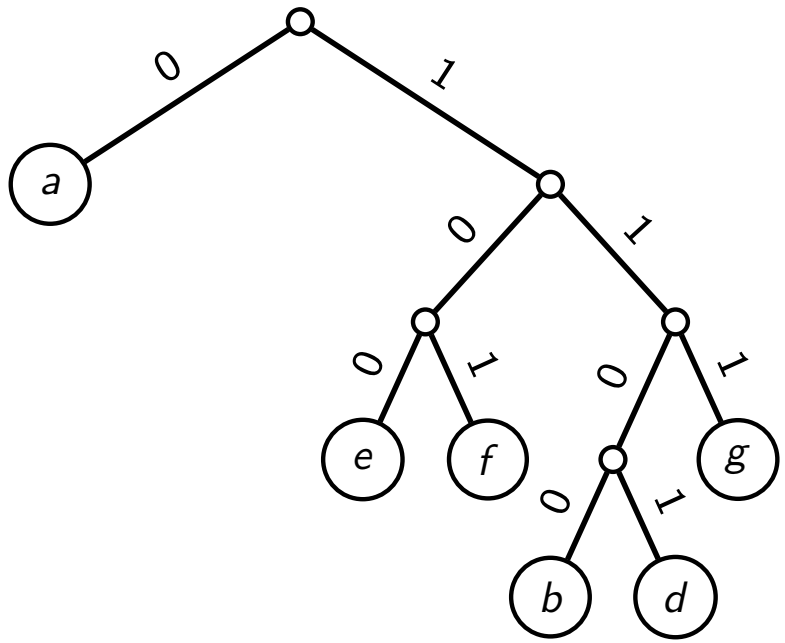
- $c(a) = 000$
- $c(b) = 001$
- $c(c) = 010$
- $c(d) = 011$
- $c(e) = 100$
- $c(f) = 101$
- $c(g) = 110$
- $c(h) = 111$



Αναπαράσταση κωδικοποιήσεων σε δυαδικά δέντρα

Κωδικοποίηση μεταβλητού μήκους

- $c(a) = 0$
- $c(e) = 100$
- $c(f) = 101$
- $c(g) = 111$
- $c(b) = 1100$
- $c(d) = 1101$



Αποκωδικοποίηση απροθεματικής κωδικοποίησης μεταβλητού μήκους

- Διαβάζουμε από αριστερά προς τα δεξιά τα δυφία και παράλληλα ακολουθούμε το αντίστοιχο μονοπάτι στο δέντρο της κωδικοποίησης από τη ρίζα προς τα φύλλα.

Αποκωδικοποίηση απροθεματικής κωδικοποίησης μεταβλητού μήκους

- Διαβάζουμε από αριστερά προς τα δεξιά τα δυφία και παράλληλα ακολουθούμε το αντίστοιχο μονοπάτι στο δέντρο της κωδικοποίησης από τη ρίζα προς τα φύλλα.

- Σύμβαση: Αν το δυφίο είναι 0 πάμε στο αριστερό παιδί ενώ αν είναι 1 πάμε στο δεξιό παιδί.

Αποκωδικοποίηση απροθεματικής κωδικοποίησης μεταβλητού μήκους

- Διαβάζουμε από αριστερά προς τα δεξιά τα δυφία και παράλληλα ακολουθούμε το αντίστοιχο μονοπάτι στο δέντρο της κωδικοποίησης από τη ρίζα προς τα φύλλα.

- Σύμβαση: Αν το δυφίο είναι 0 πάμε στο αριστερό παιδί ενώ αν είναι 1 πάμε στο δεξιό παιδί.

- Όταν φτάσουμε σε φύλλο η ακολουθία των δυφίων που διαβάσαμε αντιστοιχεί στον χαρακτήρα του φύλλου.

Αποκωδικοποίηση απροθεματικής κωδικοποίησης μεταβλητού μήκους

- Διαβάζουμε από αριστερά προς τα δεξιά τα δυφία και παράλληλα ακολουθούμε το αντίστοιχο μονοπάτι στο δέντρο της κωδικοποίησης από τη ρίζα προς τα φύλλα.
- Σύμβαση: Αν το δυφίο είναι 0 πάμε στο αριστερό παιδί ενώ αν είναι 1 πάμε στο δεξιό παιδί.
- Όταν φτάσουμε σε φύλλο η ακολουθία των δυφίων που διαβάσαμε αντιστοιχεί στον χαρακτήρα του φύλλου.
- Αρχίζουμε από την αρχή για να βρούμε τον επόμενο χαρακτήρα.

Κωδικοποίηση

Άπληστα: Χαρακτήρες με μικρές συχνότητες τοποθετούνται σε φύλλα μακριά από τη ρίζα (μεγαλύτερο μήκος κωδικοποίησης) και χαρακτήρες με μεγάλες συχνότητες τοποθετούνται σε φύλλα κοντά στη ρίζα (μικρότερο μήκος κωδικοποίησης).

Αλγόριθμος

Ταξινομούμε τις συχνότητες εμφανίσεων των χαρακτήρων σε αύξουσα σειρά.

Αλγόριθμος

Ταξινομούμε τις συχνότητες εμφανίσεων των χαρακτήρων σε αύξουσα σειρά.

Αρχικοποιούμε ένα γράφημα με n κορυφές (που τις αντιστοιχίζουμε στους χαρακτήρες) και 0 ακμές

Αλγόριθμος

Ταξινομούμε τις συχνότητες εμφανίσεων των χαρακτήρων σε αύξουσα σειρά.

Αρχικοποιούμε ένα γράφημα με n κορυφές (που τις αντιστοιχίζουμε στους χαρακτήρες) και 0 ακμές

Παίρνουμε τις κορυφές που αντιστοιχούν στους χαρακτήρες x και y με τις δύο μικρότερες συχνότητες f_1 και f_2, προσθέτουμε μία νέα κορυφή, την κάνουμε γονιό των κορυφών των y και z και τις αναθέτουμε συχνότητα $f_1 + f_2$.

Αλγόριθμος

Ταξινομούμε τις συχνότητες εμφανίσεων των χαρακτήρων σε αύξουσα σειρά.

Αρχικοποιούμε ένα γράφημα με n κορυφές (που τις αντιστοιχίζουμε στους χαρακτήρες) και 0 ακμές

Παίρνουμε τις κορυφές που αντιστοιχούν στους χαρακτήρες x και y με τις δύο μικρότερες συχνότητες f_1 και f_2, προσθέτουμε μία νέα κορυφή, την κάνουμε γονιό των κορυφών των y και z και τις αναθέτουμε συχνότητα $f_1 + f_2$.

Ταξινομούμε τώρα τις συχνότητες $f_1 + f_2, f_3, \ldots, f_n$ σε αύξουσα σειρά και επαναλαμβάνουμε τη διαδικασία.

Αλγόριθμος

Ταξινομούμε τις συχνότητες εμφανίσεων των χαρακτήρων σε αύξουσα σειρά.

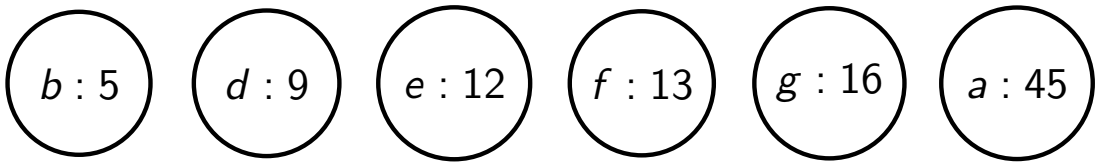
Αρχικοποιούμε ένα γράφημα με n κορυφές (που τις αντιστοιχίζουμε στους χαρακτήρες) και 0 ακμές

Παίρνουμε τις κορυφές που αντιστοιχούν στους χαρακτήρες x και y με τις δύο μικρότερες συχνότητες f_1 και f_2, προσθέτουμε μία νέα κορυφή, την κάνουμε γονιό των κορυφών των y και z και τις αναθέτουμε συχνότητα $f_1 + f_2$.

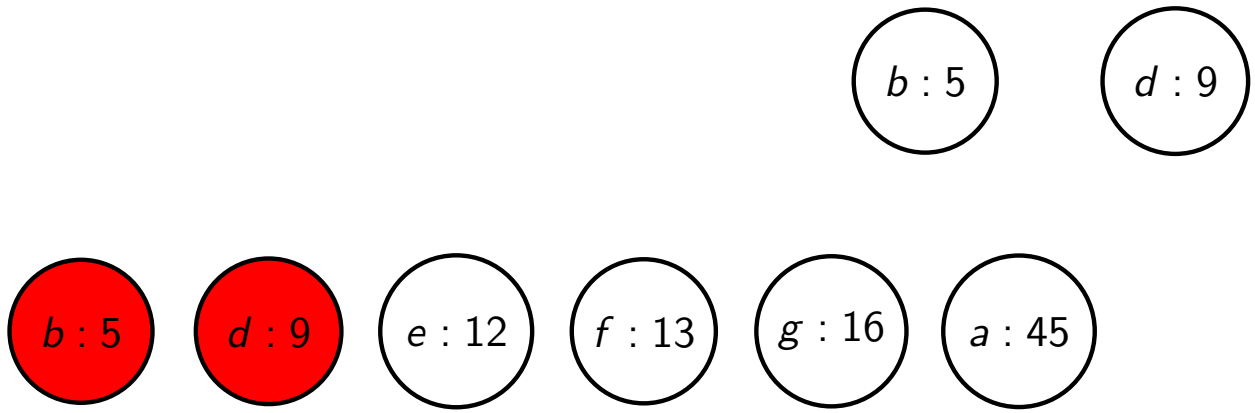
Ταξινομούμε τώρα τις συχνότητες $f_1 + f_2, f_3, \ldots, f_n$ σε αύξουσα σειρά και επαναλαμβάνουμε τη διαδικασία.

Ο αλγόριθμος τερματίζει όταν μείνει μόνο μία κορυφή με συχνότητα 100%.

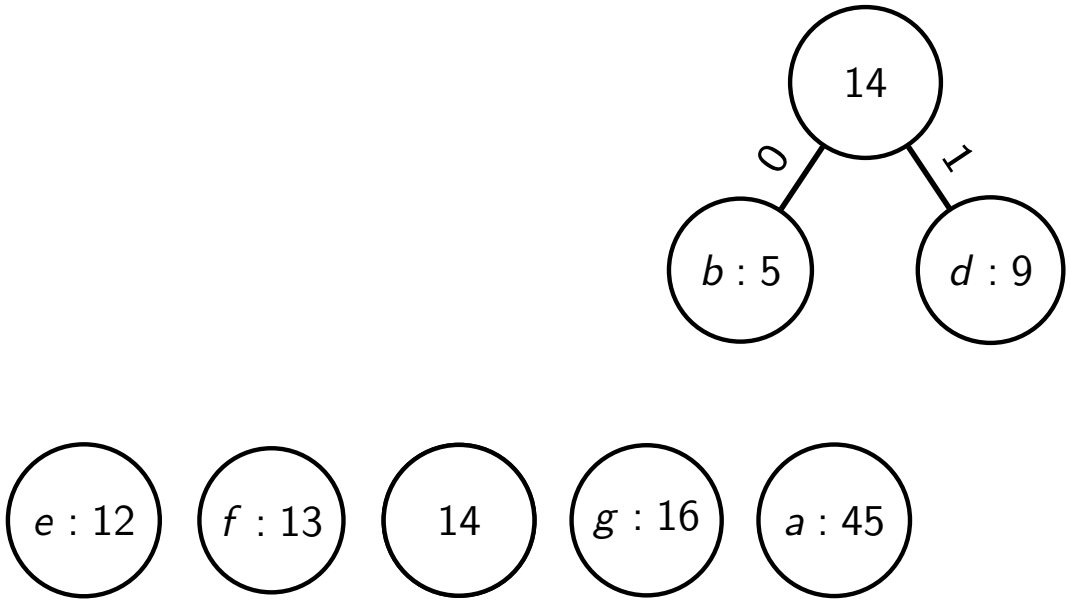
Παράδειγμα



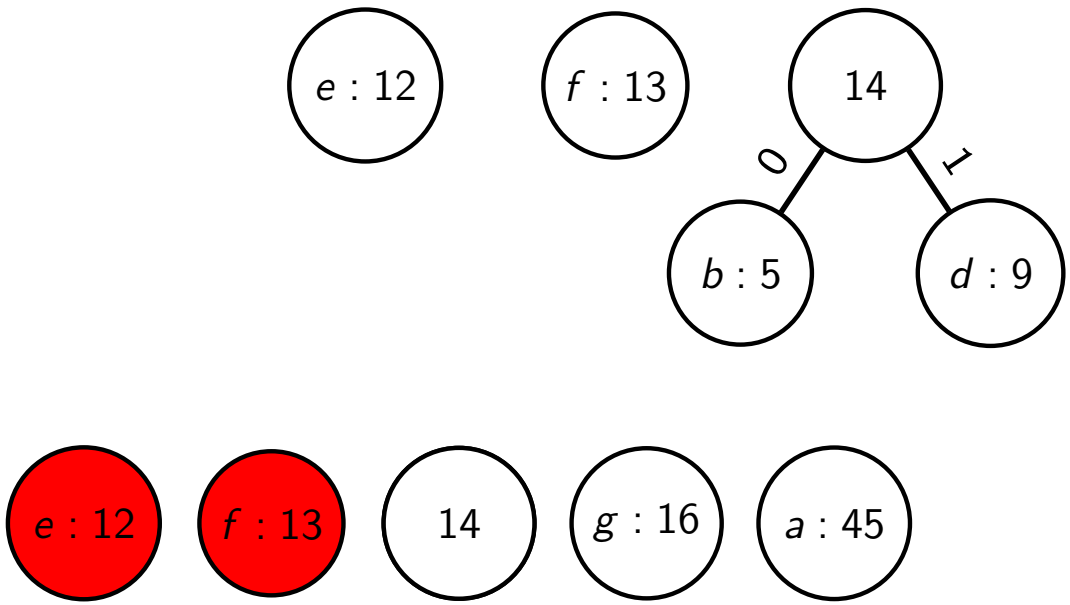
Παράδειγμα



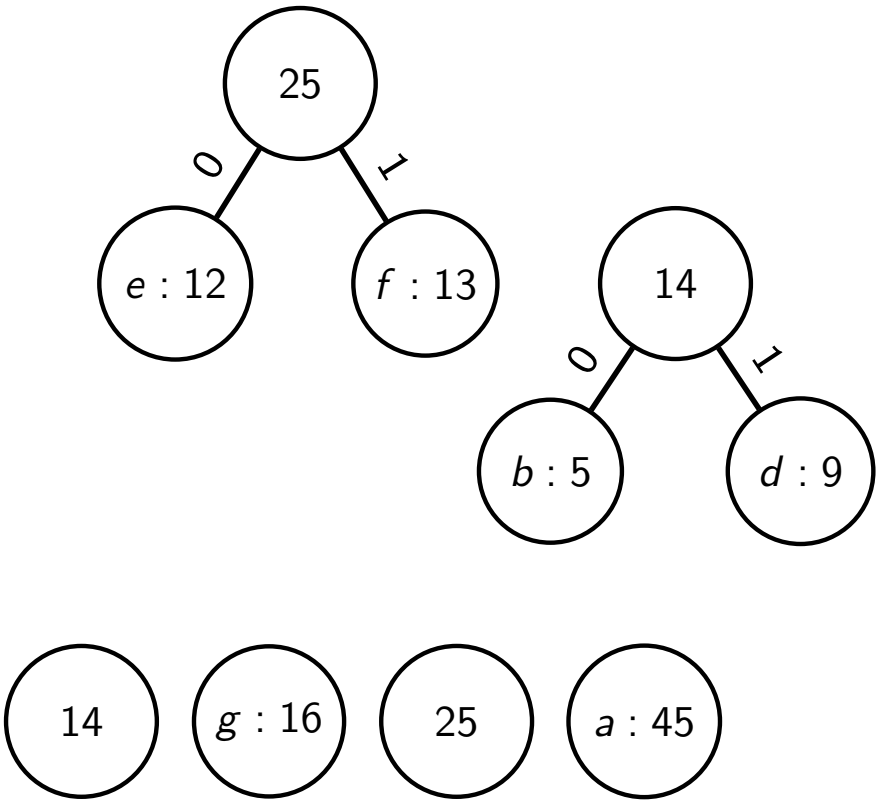
Παράδειγμα



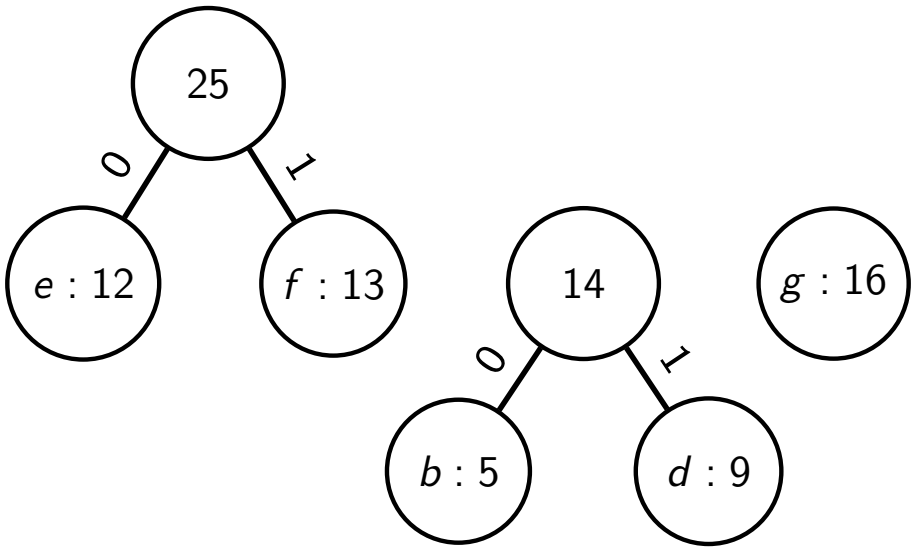
Παράδειγμα

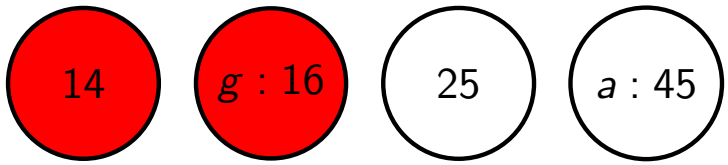


Παράδειγμα

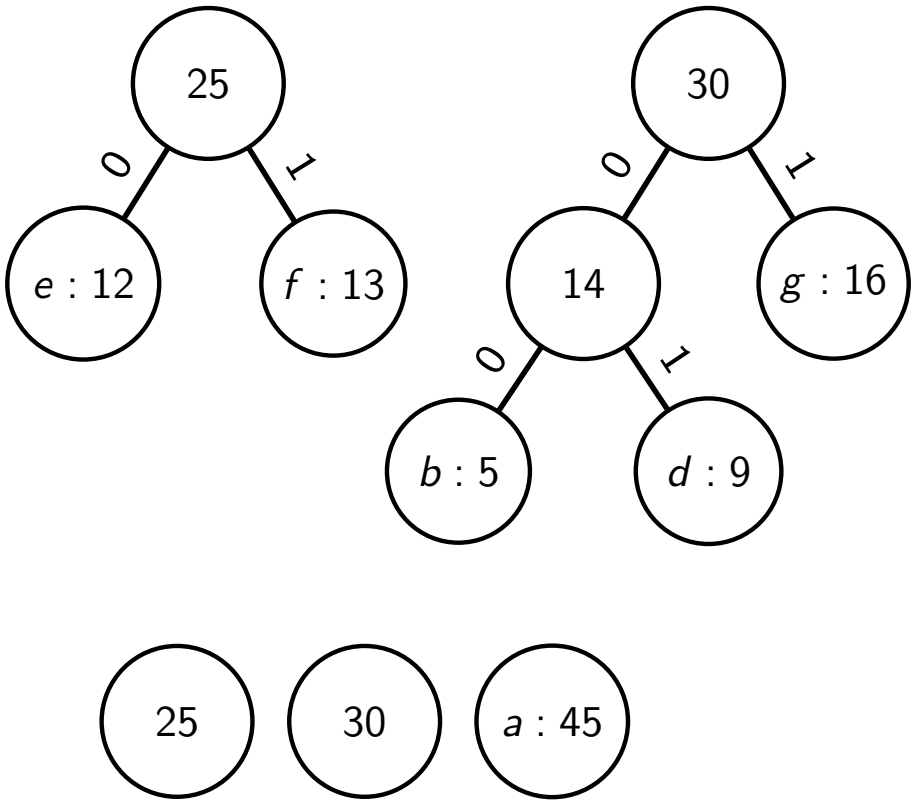


Παράδειγμα

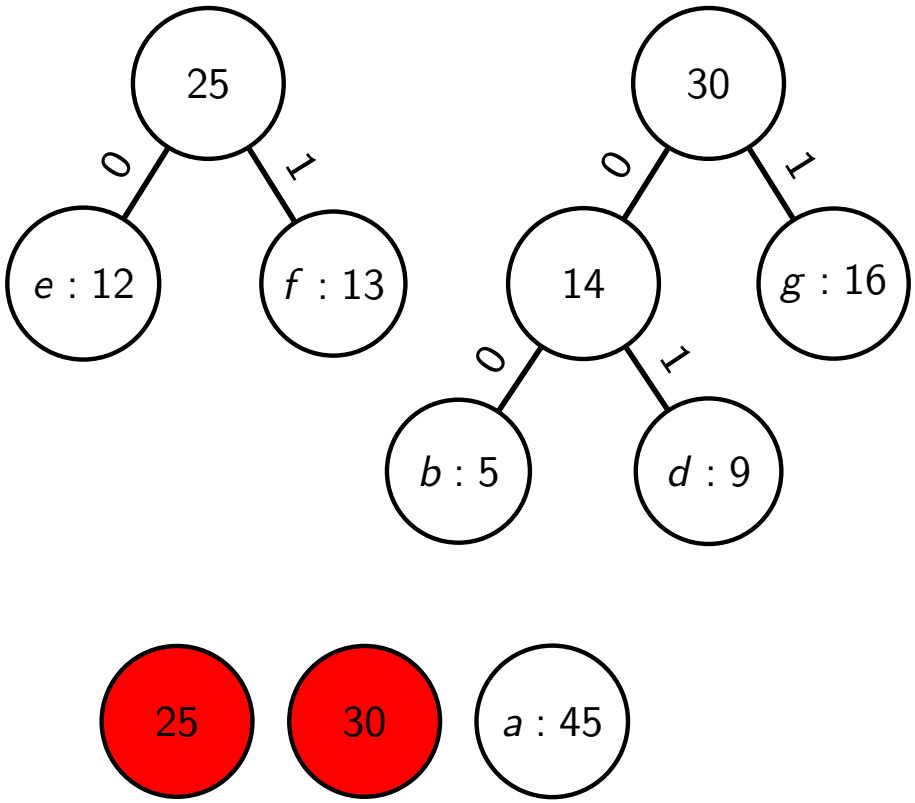




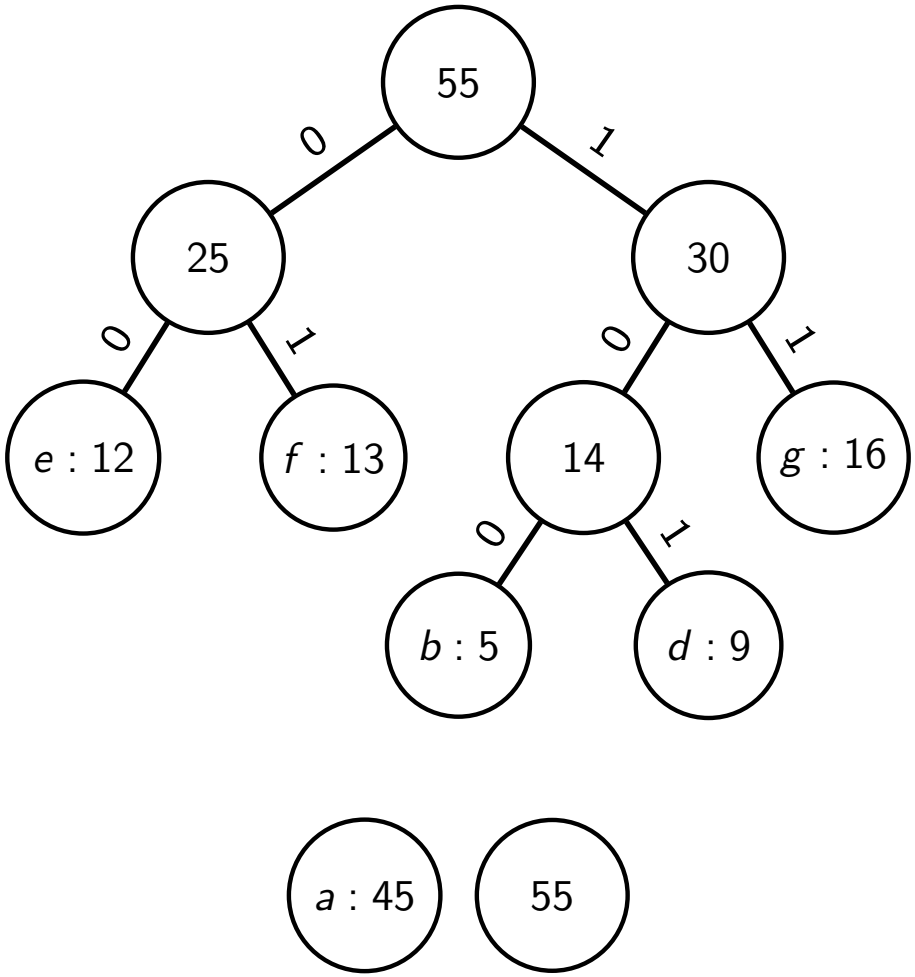
Παράδειγμα



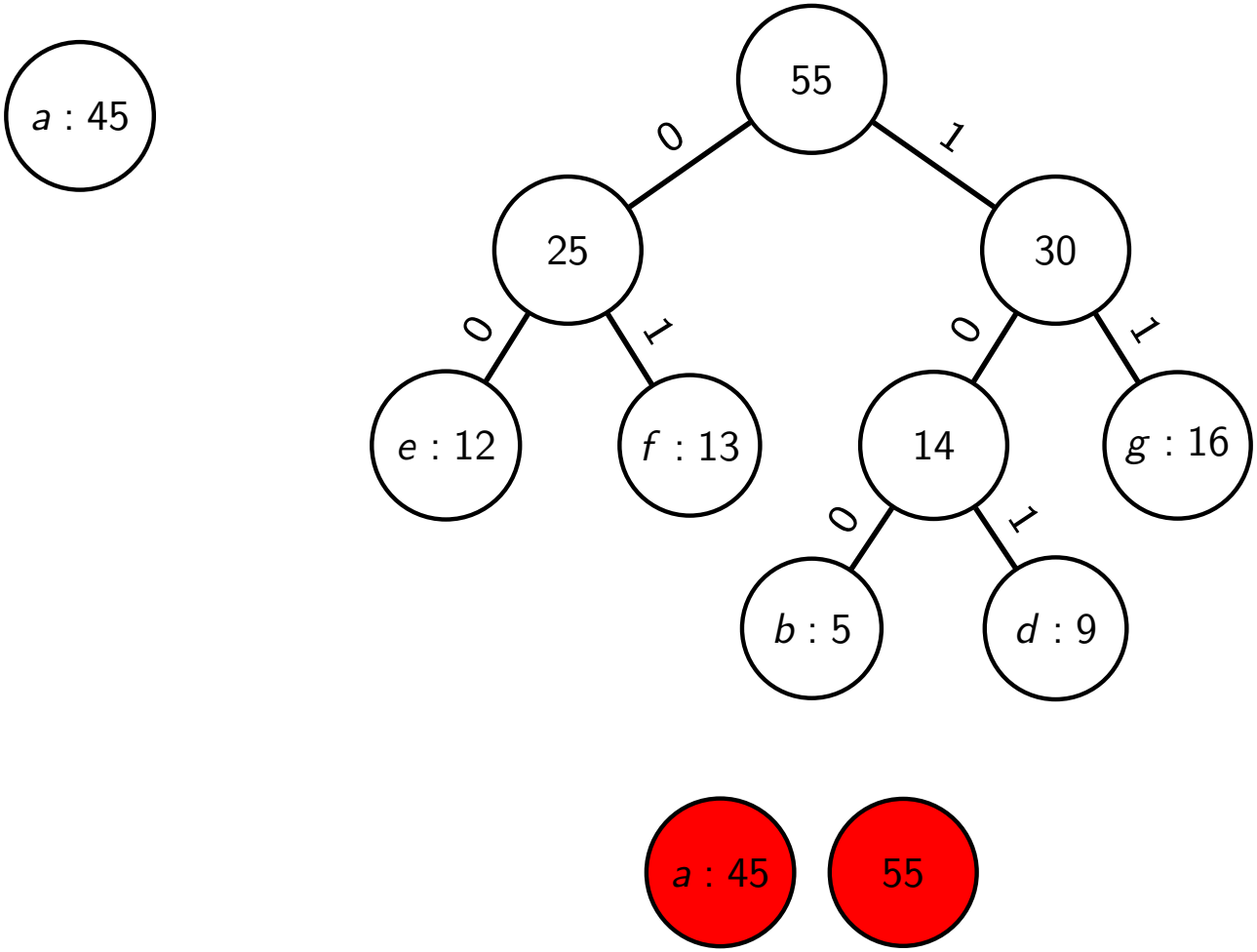
Παράδειγμα



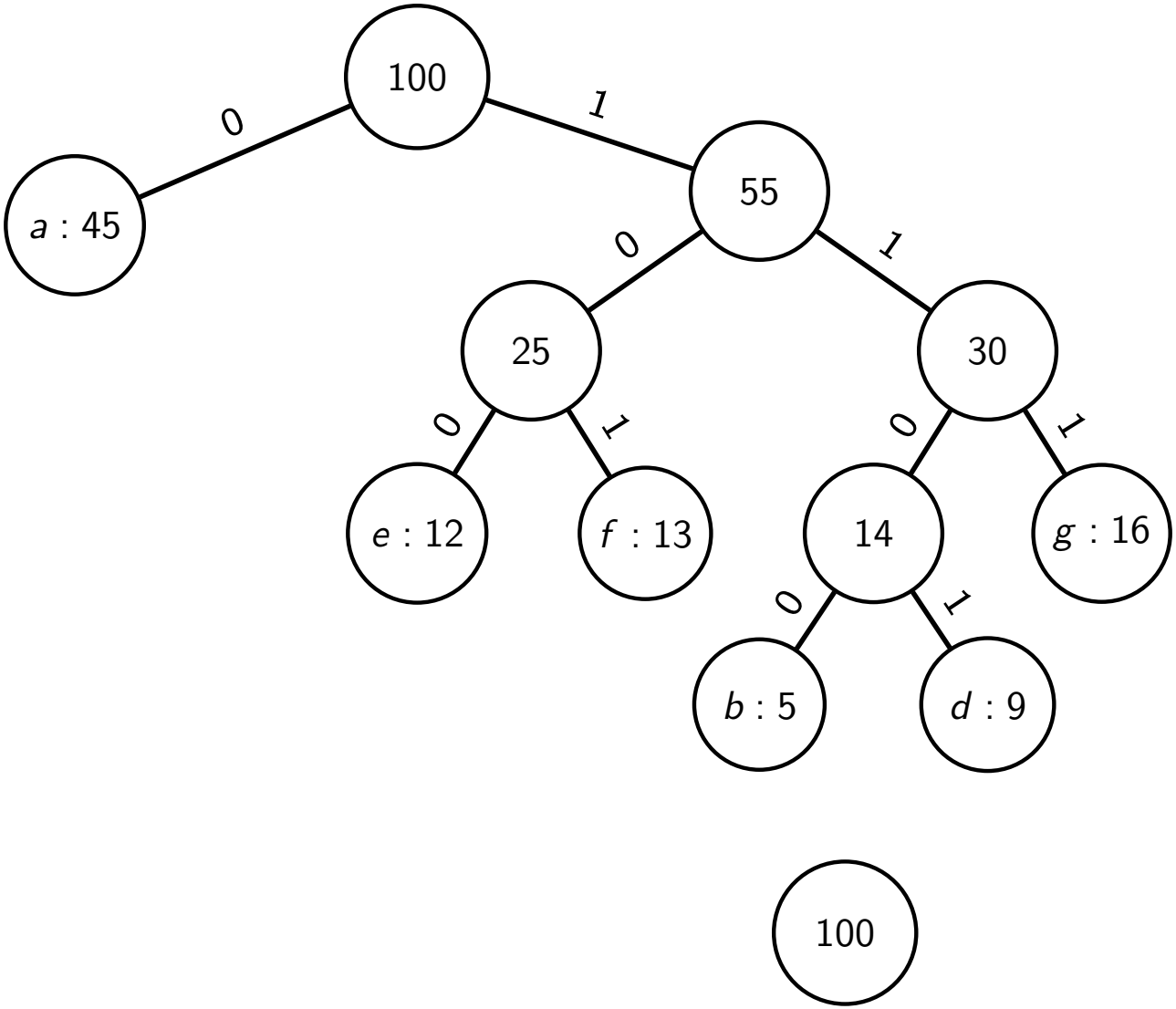
Παράδειγμα



Παράδειγμα



Παράδειγμα



100

Αλγόριθμος

Είσοδος: Ένας πίνακας S με τους n χαρακτήρες και τις συχνότητες τους
Έξοδος: Μία δεντρική αναπαράσταση της κωδικοποίησης με n φύλλα

Huffman(S)
Εάν $n = 2$
    Επίστρεψε ένα δέντρο με ρίζα και δύο φύλλα.
Αλλιώς
    Για y και z τους χαρακτήρες με την μικρότερη συχνότητα στο S
    $S' \leftarrow S$
    Αφαίρεσε τις y και z από το S'
    Εισήγαγε νέο χαρακτήρα ω στο S' με $f_\omega = f_y + f_z$
    $T' \leftarrow$ **Huffman**(S')
    $T \leftarrow T'$ μαζί με δύο παιδιά της ω που αντιστοιχούν στις y και z.
    Επίστρεψε T.

Αλγόριθμος

Είσοδος: Ένας πίνακας S με τους n χαρακτήρες και τις συχνότητες τους
Έξοδος: Μία δεντρική αναπαράσταση της κωδικοποίησης με n φύλλα

Huffman(S)
Εάν $n = 2$
  Επίστρεψε ένα δέντρο με ρίζα και δύο φύλλα.
Αλλιώς
  Για y και z τους χαρακτήρες με την μικρότερη συχνότητα στο S
  $S' \leftarrow S$
  Αφαίρεσε τις y και z από το S'
  Εισήγαγε νέο χαρακτήρα ω στο S' με $f_\omega = f_y + f_z$
  $T' \leftarrow$ **Huffman**(S')
  $T \leftarrow T'$ μαζί με δύο παιδιά της ω που αντιστοιχούν στις y και z.
  Επίστρεψε T.

Πολυπλοκότητα: Ο αλγόριθμος θα εκτελέσει $\mathcal{O}(n \log n)$ βήματα επειδή για την χρονική του πολυπλοκότητα ισχύει ότι $T(n) = T(n-1) + \mathcal{O}(\log n)$.

Ασκήσεις

Δείξτε ότι το δέντρο Huffman κωδικοποίησης n χαρακτήρων έχει $2n - 1$ κορυφές (n φύλλα και $n - 1$ εσωτερικές κορυφές).

Δείξτε ότι η συχνότητα της ρίζας είναι $N = \sum_{x \in S} f_x$.

Ορθότητα Αλγόριθμου - Παρατηρήσεις

Θεώρημα: Σε ένα βέλτιστο δέντρο T αν ℓ_k και ℓ_j είναι τα φύλλα που αντιστοιχούν στους χαρακτήρες k και j με συχνότητες $f(k) \leq f(j)$ ισχύει ότι $\mathbf{d}_T(\ell_k) \geq \mathbf{d}_T(\ell_j)$, όπου $\mathbf{d}_T(\ell_t)$ είναι το βάθος του φύλλου στο οποίο αντιστοιχεί ο χαρακτήρας t στο T.

Ισοδύναμα, το βάθος των φύλλων των χαρακτήρων στο δέντρο είναι αντιστρόφως ανάλογο των συχνοτήτων τους.

Ορθότητα Αλγόριθμου - Παρατηρήσεις

Έστω $L(C)$ το μήκος μίας βέλτιστης κωδικοποίησης με $f(k) < f(j)$ και $\mathbf{d}_T(\ell_k) < \mathbf{d}_T(\ell j)$. Ανταλλάσσουμε τα φύλλα στα οποία αντιστοιχούν οι χαρακτήρες k και j και προκύπτει μία κωδικοποίηση $L(C')$, με $L(C') \leq L(C)$.

Τότε

$$L(C) = \sum_{x \in S \setminus \{k,j\}} f(x)\mathbf{d}_T(\ell_x) + f(k)\mathbf{d}_T(\ell_k) + f(j)\mathbf{d}_T(\ell_j)$$

και

$$L(C') = \sum_{x \in S \setminus \{k,j\}} f(x)\mathbf{d}_T(\ell_x) + f(k)\mathbf{d}_T(\ell_j) + f(j)\mathbf{d}_T(\ell_k)$$

$$\begin{aligned} L(C) - L(C') &= f(k)\mathbf{d}_T(\ell_k) + f(j)\mathbf{d}_T(\ell_j) - f(k)\mathbf{d}_T(\ell_j) - f(j)\mathbf{d}_T(\ell_k) \\ &= f(k)(\mathbf{d}_T(\ell_k) - \mathbf{d}_T(\ell_j)) + f(j)(\mathbf{d}_T(\ell_j) - \mathbf{d}_T(\ell_k)) \\ &= (f(k) - f(j))(\mathbf{d}_T(\ell_k) - \mathbf{d}_T(\ell_j)) > 0 \end{aligned}$$

Άτοπο!

Ορθότητα Αλγόριθμου - Παρατηρήσεις

Θεώρημα: Σε κάθε βέλτιστο δέντρο κάθε εσωτερική κορυφή έχει ακριβώς 2 παιδιά.

Οι δύο μεγαλύτερες σε μήκος κωδικοποιήσεις έχουν το ίδιο μήκος.

Ορθότητα Αλγόριθμου - Παρατηρήσεις

Θεώρημα: Σε κάθε βέλτιστο δέντρο κάθε εσωτερική κορυφή έχει ακριβώς 2 παιδιά.

Οι δύο μεγαλύτερες σε μήκος κωδικοποιήσεις έχουν το ίδιο μήκος.

Αν το δέντρο περιέχει μία κορυφή v με ένα παιδί u συνθλίβουμε την ακμή $\{v, u\}$ ώστε να ενωθούν οι κορυφές v και u καταλήγοντας σε ένα δέντρο που αντιστοιχεί σε μία κωδικοποίηση των χαρακτήρων με μικρότερο κόστος.

Ορθότητα Αλγόριθμου - Παρατηρήσεις

Θεώρημα: Σε κάθε βέλτιστο δέντρο κάθε εσωτερική κορυφή έχει ακριβώς 2 παιδιά.

Οι δύο μεγαλύτερες σε μήκος κωδικοποιήσεις έχουν το ίδιο μήκος.

Αν το δέντρο περιέχει μία κορυφή v με ένα παιδί u συνθλίβουμε την ακμή $\{v, u\}$ ώστε να ενωθούν οι κορυφές v και u καταλήγοντας σε ένα δέντρο που αντιστοιχεί σε μία κωδικοποίηση των χαρακτήρων με μικρότερο κόστος.

Δύο από τις μεγαλύτερες σε μήκος κωδικοποιήσεις αντιστοιχούν στους δύο χαρακτήρες με τις δύο μικρότερες συχνότητες και οι κωδικοποιήσεις τους διαφέρουν μόνο στο τελευταίο δυφίο.

Ορθότητα Αλγόριθμου - Παρατηρήσεις

$$\textbf{cost}(T') = \textbf{cost}(T) - f_\omega$$

Όπου T και T' τα δέντρα που φτιάχνει ο αλγόριθμος παραπάνω.
(Το T περιέχει τις y και z ενώ το T' περιέχει την ω.)

Ορθότητα Αλγόριθμου - Παρατηρήσεις

$$\mathbf{cost}(T') = \mathbf{cost}(T) - f_\omega$$

Όπου T και T' τα δέντρα που φτιάχνει ο αλγόριθμος παραπάνω.
(Το T περιέχει τις y και z ενώ το T' περιέχει την ω.)

$$\begin{aligned}
\mathbf{cost}(T) &= \sum_{x \in S} f_x \mathbf{d}_T(\ell_x) \\
&= \sum_{x \in S \setminus \{y,z\}} f_x \mathbf{d}_T(\ell_x) + f_y \mathbf{d}_T(\ell_y) + f_z \mathbf{d}_T(\ell_z) \\
&= \sum_{x \in S \setminus \{y,z\}} f_x \mathbf{d}_T(\ell_x) + (f_y + f_z)(1 + \mathbf{d}_T(\ell_\omega)) \\
&= \sum_{x \in S \setminus \{y,z\}} f_x \mathbf{d}_T(\ell_x) + (f_\omega)(1 + \mathbf{d}_T(\ell_\omega)) \\
&= \sum_{x \in S'} f_x \mathbf{d}_T(\ell_x) + f_\omega \\
&= \mathbf{cost}(T') + f_\omega.
\end{aligned}$$

Ορθότητα Αλγόριθμου

Θεώρημα: Για κάθε πλήθος χαρακτήρων n και κάθε συνάρτηση συχνότητας αυτών, το δέντρο/κωδικοποίηση που παράγει ο αλγόριθμος Huffman αποτελεί βέλτιστη κωδικοποίηση.

Ορθότητα Αλγόριθμου

Θεώρημα: Για κάθε πλήθος χαρακτήρων n και κάθε συνάρτηση συχνότητας αυτών, το δέντρο/κωδικοποίηση που παράγει ο αλγόριθμος Huffman αποτελεί βέλτιστη κωδικοποίηση.

Απόδειξη με επαγωγή στο μέγεθος του S.

- Για $k = 2$ το δυαδικό δέντρο με μία ρίζα και δύο παιδιά είναι πάντα βέλτιστο.

Ορθότητα Αλγόριθμου

Θεώρημα: Για κάθε πλήθος χαρακτήρων n και κάθε συνάρτηση συχνότητας αυτών, το δέντρο/κωδικοποίηση που παράγει ο αλγόριθμος Huffman αποτελεί βέλτιστη κωδικοποίηση.

Απόδειξη με επαγωγή στο μέγεθος του S.

- Για $k = 2$ το δυαδικό δέντρο με μία ρίζα και δύο παιδιά είναι πάντα βέλτιστο.

- Έστω ότι για $k = n - 1$ και κάθε συνάρτηση συχνοτήτων f το δέντρο που φτιάχνει ο αλγόριθμος είναι το βέλτιστο.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\textbf{cost}(Z) < \textbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\mathbf{cost}(Z) < \mathbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.
  - Το Z' παράγεται από το Z αφαιρώντας τις x και y, ονομάζοντας τον κοινό γονιό τους ω και δίνοντας του συχνότητα $f_x + f_y$.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\mathbf{cost}(Z) < \mathbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.
  - Το Z' παράγεται από το Z αφαιρώντας τις x και y, ονομάζοντας τον κοινό γονιό τους ω και δίνοντας του συχνότητα $f_x + f_y$.
  - Αντίστοιχα ορίζουμε το T' από το δέντρο T που κατασκεύασε ο αλγόριθμος.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\mathbf{cost}(Z) < \mathbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.
  - Το Z' παράγεται από το Z αφαιρώντας τις x και y, ονομάζοντας τον κοινό γονιό τους ω και δίνοντας του συχνότητα $f_x + f_y$.
  - Αντίστοιχα ορίζουμε το T' από το δέντρο T που κατασκεύασε ο αλγόριθμος.
  - $\mathbf{cost}(Z') = \mathbf{cost}(Z) - f_\omega$ και $\mathbf{cost}(T') = \mathbf{cost}(T) - f_\omega$.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\textbf{cost}(Z) < \textbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.
  - Το Z' παράγεται από το Z αφαιρώντας τις x και y, ονομάζοντας τον κοινό γονιό τους ω και δίνοντας του συχνότητα $f_x + f_y$.
  - Αντίστοιχα ορίζουμε το T' από το δέντρο T που κατασκεύασε ο αλγόριθμος.
  - $\textbf{cost}(Z') = \textbf{cost}(Z) - f_\omega$ και $\textbf{cost}(T') = \textbf{cost}(T) - f_\omega$.
  - Αφού $\textbf{cost}(Z) < \textbf{cost}(T)$ τότε $\textbf{cost}(Z') < \textbf{cost}(T')$.

Ορθότητα Αλγόριθμου

- Έστω $k = n$ και έστω ότι το δέντρο T που παράγει ο αλγόριθμος δεν είναι βέλτιστο.
  - Υπάρχει ένα δέντρο Z με $\textbf{cost}(Z) < \textbf{cost}(T)$ και οι κορυφές x και y με τις δύο μικρότερες συχνότητες είναι αδέρφια.
  - Το Z' παράγεται από το Z αφαιρώντας τις x και y, ονομάζοντας τον κοινό γονιό τους ω και δίνοντας του συχνότητα $f_x + f_y$.
  - Αντίστοιχα ορίζουμε το T' από το δέντρο T που κατασκεύασε ο αλγόριθμος.
  - $\textbf{cost}(Z') = \textbf{cost}(Z) - f_\omega$ και $\textbf{cost}(T') = \textbf{cost}(T) - f_\omega$.
  - Αφού $\textbf{cost}(Z) < \textbf{cost}(T)$ τότε $\textbf{cost}(Z') < \textbf{cost}(T')$.
  - Άτοπο γιατί από την επαγωγική υπόθεση ο αλγόριθμος υπολογίζει κωδικοποίηση βέλτιστου κόστους για κάθε σύνολο $n-1$ χαρακτήρων και συχνοτήτων.

Άσκηση

Ποιά/ποιές από τις παρακάτω κωδικοποιήσεις αποτελεί/ουν ορθη κωδικοποίηση Huffman της λέξης abracadabra;

α′) $r = 000, c = 0010, d = 0011, b = 01, a = 1$

β′) $r = 000, c = 0010, d = 0011, b = 11, a = 1$

γ′) $r = 110, c = 1110, d = 1111, b = 10, a = 0$

δ′) $a = 110, c = 1110, d = 1111, b = 10, r = 0$

Άσκηση

Ένα τέλειο ταίριασμα σε ένα γράφημα είναι ένα σύνολο από ακμές έτσι ώστε κάθε κορυφή να περιέχεται σε ακριβώς μία ακμή. Για παράδειγμα, το δέντρο στα αριστερά έχει τέλειο ταίριασμα αλλά το δέντρο στα δεξιά δεν έχει.

Περιγράψτε σε φυσική γλώσσα έναν άπληστο αλγόριθμο που να αποφασίζει εάν ένα δέντρο έχει τέλειο ταίριασμα ή όχι και αιτιολογήστε με 1-2 προτάσεις την ορθότητά του. (Με άλλα λόγια, αιτιολογήστε γιατί επιλέξατε το συγκεκριμένο κριτήριο άπληστης επιλογής)