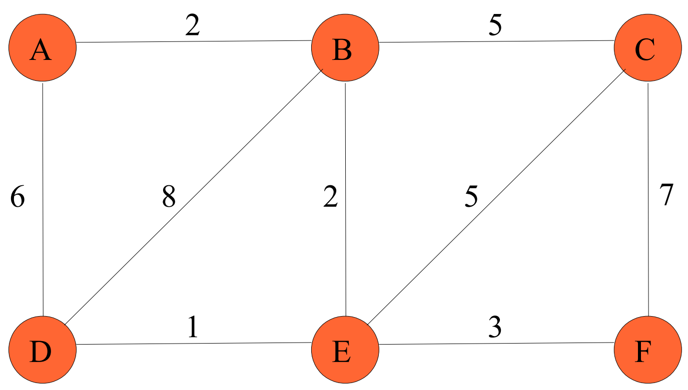
Αλγόριθμοι και Πολυπλοκότητα
3ο Σύνολο Ασκήσεων

Ονομ/νυμο: Βασίλειος-Ιωάννης Μανάβης
Α.Μ.: 1115201000136

1.
Το τελευταίο ψηφίο του αριθμού του τηλεφώνου μου είναι το 1, το προτελευταίο είναι το 6 και το αντιπροτελευταίο είναι 3. Άρα x=1, y=6 και z=3. Επομένως ο γράφος μπορεί να αναπαρασταθεί ως εξής:



1. **<u>Βήμα 1:</u>**
   Ας ξεκινήσουμε από τον κόμβο A.
   Σχήμα 1.1.1
   <u>Βήμα 2:</u>
   near[B] = A, dist[B] = 2
   near[D] = A, dist[D] = 6
   Άρα θα επισκεφτούμε τον κόμβο B μέσω του A.
   Σχήμα 1.1.2
   <u>Βήμα 3:</u>
   near[D] = A, dist[D] = 6
   near[C] = B, dist[C] = 5
   near[D] = B, dist[D] = 8
   near[E] = B, dist[E] = 2
   Άρα θα επισκεφτούμε τον κόμβο E μέσω του B.
   Σχήμα 1.1.3
   <u>Βήμα 4:</u>
   near[C] = B, dist[C] = 5
   near[D] = B, dist[D] = 8
   near[C] = E, dist[C] = 5
   near[D] = E, dist[D] = 1
   near[F] = E, dist[F] = 3
   Άρα θα επισκεφτούμε τον κόμβο D μέσω του E.
   Σχήμα 1.1.4
   <u>Βήμα 5:</u>
   near[C] = E, dist[C] = 5
   near[F] = E, dist[F] = 3
   near[D] = D, dist[D] = 0
   Άρα θα επισκεφτούμε τον κόμβο F μέσω του E.
   Σχήμα 1.1.5

Βήμα 6:
near[C] = E, dist[C] = 5
near[C] = F, dist[C] = 7
Άρα θα επισκεφτούμε τον κόμβο C μέσω του E.
Σχήμα 1.1.6

2. Διάταξη ακμών σε αύξουσα σειρά
(D,E): κόστος 1
(A,B): κόστος 2
(B,E): κόστος 2
(E,F): κόστος 3
(B,C): κόστος 5
(C,E): κόστος 5
(A,D): κόστος 6
(C,F): κόστος 7
(B,D): κόστος 8

Βήμα 1:
Προσθέτουμε την πρώτη ακμή που βρίσκουμε στη σειρά που φτιάξαμε παραπάνω. Αυτή περιέχει τους κόμβους D και E.
find(D) = D
find(E) = E
Union(D,E)=1
find(D) ≠ find(E) Άρα, δεν έχουμε κύκλο.
Σχήμα 1.2.1
Βήμα 2:
Προσθέτουμε τους κόμβους D και E στο δέντρο επικάλυψης. Προσθέτουμε την επόμενη ακμή στη σειρά. Αυτή περιέχει τους κόμβους A και B.
find(A) = A
find(B) = B
Union(A,B)=2
find(A) ≠ find(B) Άρα, δεν έχουμε κύκλο.
Σχήμα 1.2.2
Βήμα 3:
Προσθέτουμε τους κόμβους A και B στο δέντρο επικάλυψης. Προσθέτουμε την επόμενη ακμή στη σειρά. Αυτή περιέχει τους κόμβους B και E.
find(B) = B
find(E) = E
Union(B,E)=2
find(B) ≠ find(E) Άρα, δεν έχουμε κύκλο.
Σχήμα 1.2.3
Βήμα 4:
Οι κόμβοι B και E υπάρχουν ήδη στο δέντρο. Προσθέτουμε την επόμενη ακμή στη σειρά. Αυτή περιέχει τους κόμβους E και F.
find(E) = E
find(F) = F
Union(E,F)=3
find(E) ≠ find(F) Άρα, δεν έχουμε κύκλο.
Σχήμα 1.2.4
Βήμα 5:
Προσθέτουμε τον κόμβο F στο δέντρο επικάλυψης μιας και ο κόμβος E υπάρχει ήδη. Προσθέτουμε την επόμενη ακμή στη σειρά. Αυτή περιέχει τους κόμβους B και C.
find(B) = F

find(C) = C
Union(F,C)=3
find(E) ≠ find(F) Άρα, δεν έχουμε κύκλο.
Σχήμα 1.2.5

Βήμα 6:

Προσθέτουμε τον κόμβο C στο δέντρο επικάλυψης μιας και ο κόμβος B υπάρχει ήδη. Έχουμε ΔΕΕΚ.
Σχήμα 1.2.6

Βήμα 7:

Επόμενη στη σειρά είναι η πλευρά (C,E).
find(C) = C
find(E) = C
find(C) = find(E) Η πλευρά δημιουργεί κύκλο.

Βήμα 8:

Επόμενη στη σειρά είναι η πλευρά (A,D).
find(A) = C
find(E) = C
find(A) = find(E) Η πλευρά δημιουργεί κύκλο.

Βήμα 9:

Επόμενη στη σειρά είναι η πλευρά (C,F).
find(C) = C
find(F) = C
find(C) = find(F) Η πλευρά δημιουργεί κύκλο.

Βήμα 10:

Επόμενη στη σειρά είναι η πλευρά (B,D).
find(B) = C
find(D) = C
find(B) = find(D) Η πλευρά δημιουργεί κύκλο.

2.

1. Σε κάθε βήμα επιλέγουμε (σειριακά) έναν κόμβο και ενημερώνουμε τους γείτονές του. Σε κάθε βήμα "φιξάρεται" ένας κόμβος (FIXE) ορίζοντας τον πατέρα του κόμβου P (δηλαδή τον προηγούμενο κόμβο του τρέχοντος κόμβου για τον οποίο ισχύει το ελάχιστο μονοπάτι) και το κόστος της διαδρομής από τον τρέχοντα κόμβο, προς τον πατέρα (V).

Βήματα	**Δεδομένα**	**F**	**E**	**D**	**C**	**B**	**A**
Βήμα 1 Σχήμα 2.1.1	FIXE	1	0	0	0	0	0
	P	F	F	0	F	0	0
	V	0	3	+∞	7	+∞	+∞
Βήμα 2 Σχήμα 2.1.2	FIXE	1	1	0	0	0	0
	P	F	F	E	F	E	0
	V	0	3	4	7	5	+∞
Βήμα 3 Σχήμα 2.1.3	FIXE	1	1	1	0	0	0
	P	F	F	E	F	E	D
	V	0	3	4	7	5	10

Βήμα 4 Σχήμα 2.1.4	FIXE	1	1	1	1	0	0
	P	F	F	E	F	E	D
	V	0	3	4	7	5	10
Βήμα 5 Σχήμα 2.1.5	FIXE	1	1	1	1	1	0
	P	F	F	E	F	E	B
	V	0	3	4	7	5	7
Βήμα 6 Σχήμα 2.1.6	FIXE	1	1	1	1	1	1
	P	F	F	E	F	E	B
	V	0	3	4	7	5	7

2. Ο αλγόριθμος Bellman - Ford, εν αντιθέσει με εκείνον του Dijkstra, δε φιξάρει σε κάθε βήμα έναν κόμβο αλλά μπορεί να αναθεωρήσει την άποψη του για το ελάχιστο μονοπάτι κάθε κόμβου. Με έντονη γραμματοσειρά φαίνονται αυτές οι αλλαγές. Το k είναι το μήκος του μονοπατιού που εξετάζεται. Το πρώτο γράμμα είναι ο αμέσως προηγούμενος κόμβος (πατέρας) του ελάχιστου μονοπατιού από την αφετηρία F προς τον τρέχοντα κόμβο. Αμέσως δίπλα είναι το κόστος του μονοπατιού αυτού. Ο αλγόριθμος τελειώνει όταν δεν έχουμε καμία αλλαγή σε κανένα κόμβο.

Bellman - Ford						
	F	**E**	**D**	**C**	**B**	**A**
k = 0 Σχήμα 2.2.1	F, 0	F, +∞	F, +∞	F, +∞	F, +∞	F, +∞
k = 1 Σχήμα 2.2.2	F, 0	**F, 3**	F, +∞	**F, 7**	F, +∞	F, +∞
k = 2 Σχήμα 2.2.3	F, 0	F, 3	**E, 4**	F, 7	**E, 5**	F, +∞
k = 3 Σχήμα 2.2.4	F, 0	F, 3	E, 4	F, 7	E, 5	**B, 7**
k = 4 Σχήμα 2.2.5	F, 0	F, 3	E, 4	F, 7	E, 5	B, 7

3.

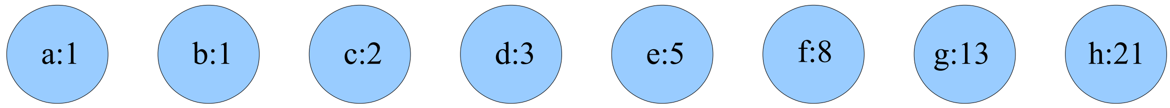
Ο αλγόριθμος Huffman είναι ο εξής:

1. Καταγραφή συμβόλων και των αντίστοιχων πιθανοτήτων. **O(n)**
2. Δημιουργείται ένας κόμβος για κάθε σύμβολο στον οποίο σημειώνεται η αντίστοιχη πιθανότητα. **O(n-1)**
3. Ακολουθεί εύρεση των δύο μικρότερων κόμβων (O(logn) για κάθε αναζήτηση, άρα **O(logn)**) οι οποίοι δεν έχουν κόμβο-πατέρα, και στη συνέχεια δημιουργείται ένας νέος διακλαδιζόμενος κόμβος στον οποίο σημειώνεται το άθροισμα των πιθανοτήτων που έχουν οι δύο κόμβοι-παιδιά (**O(1)**).
4. Επαναλαμβάνεται το 3ο βήμα μέχρι όλοι οι κόμβοι εκτός από τη ρίζα να έχουν κόμβο-πατέρα.
5. Σημειώνεται με 0 και 1 κάθε ζεύγος ακμών (0 για το αριστερό παιδί, 1 για το δεξί). **O(n)**

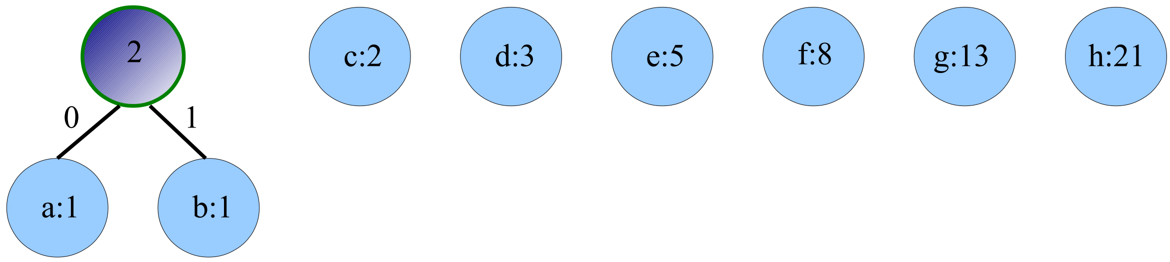
Επομένως η πολυπλοκότητα του αλγορίθμου είναι **O(nlogn)**.

1. A:1, b:1, c:2, d:3, e:5, f:8, g:13, h:21

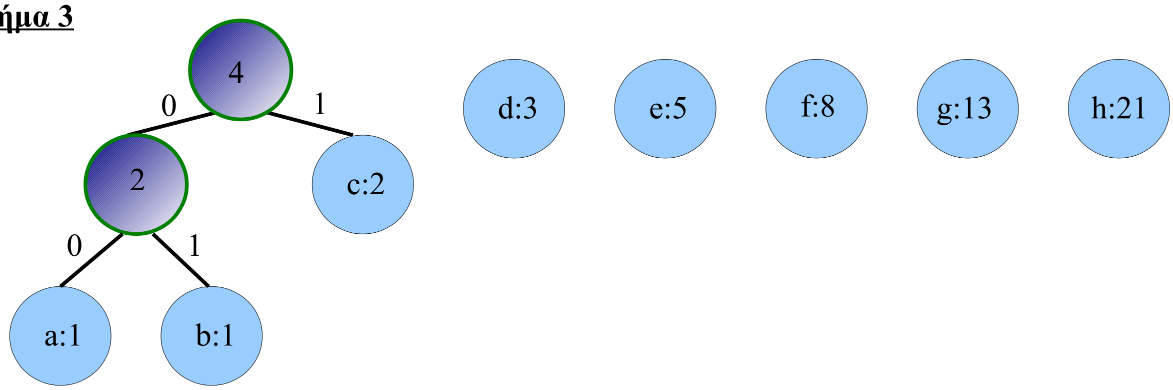
Βήμα 1



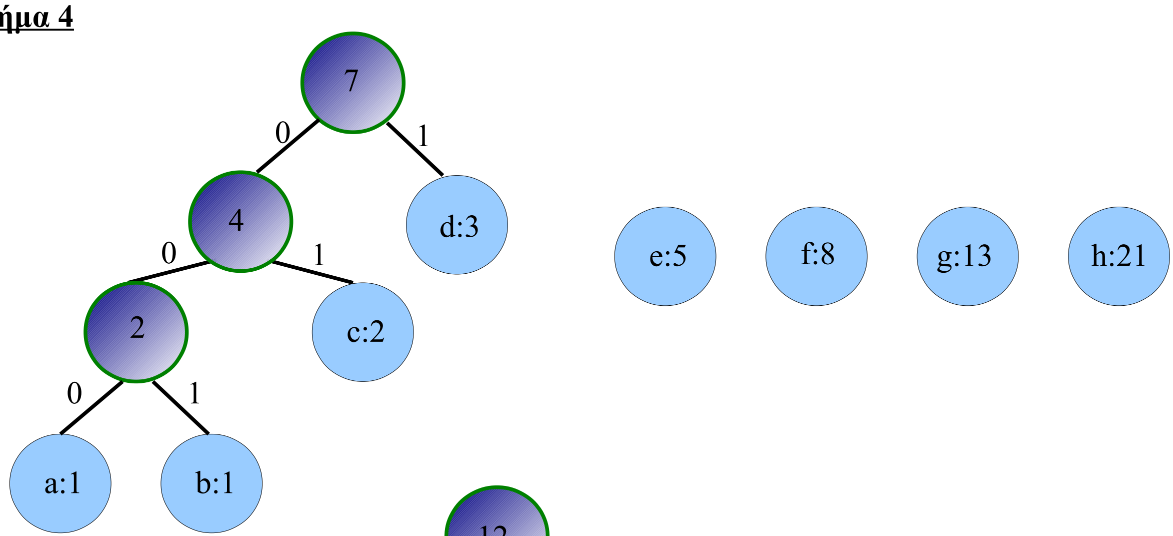
Βήμα 2



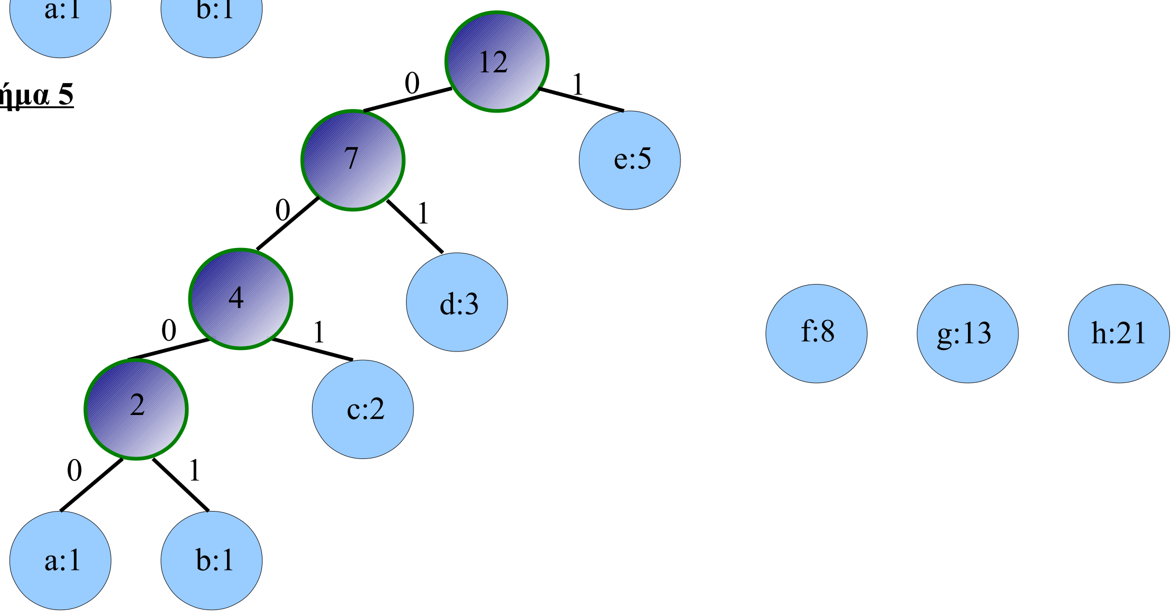
Βήμα 3



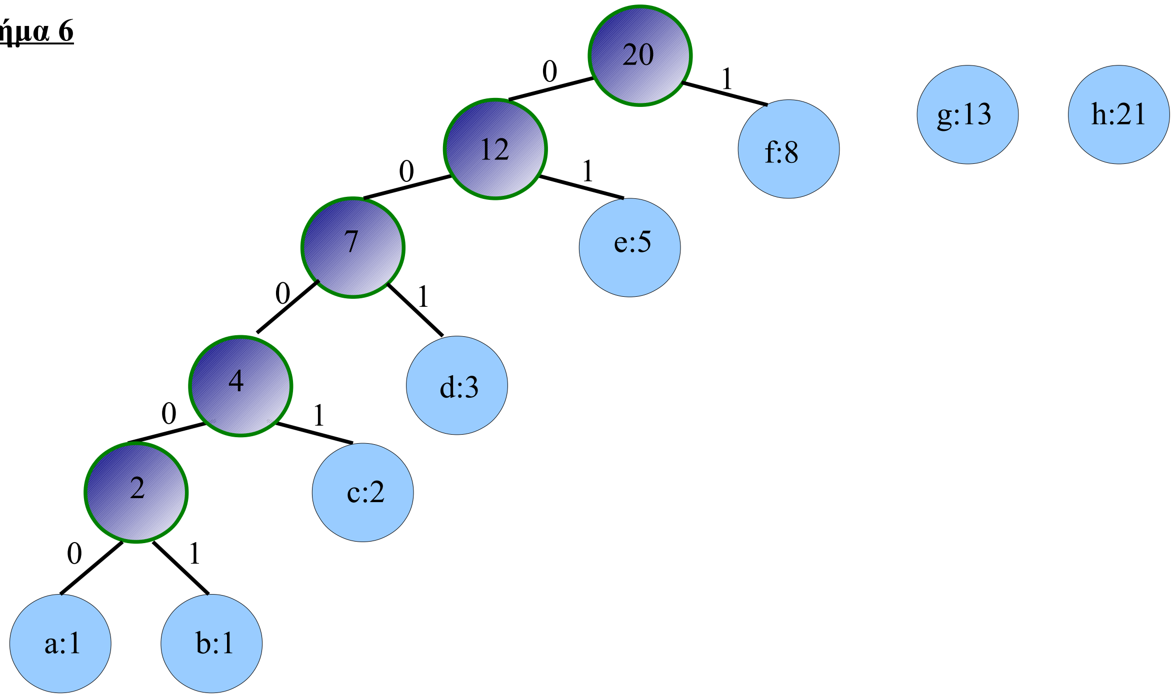
Βήμα 4



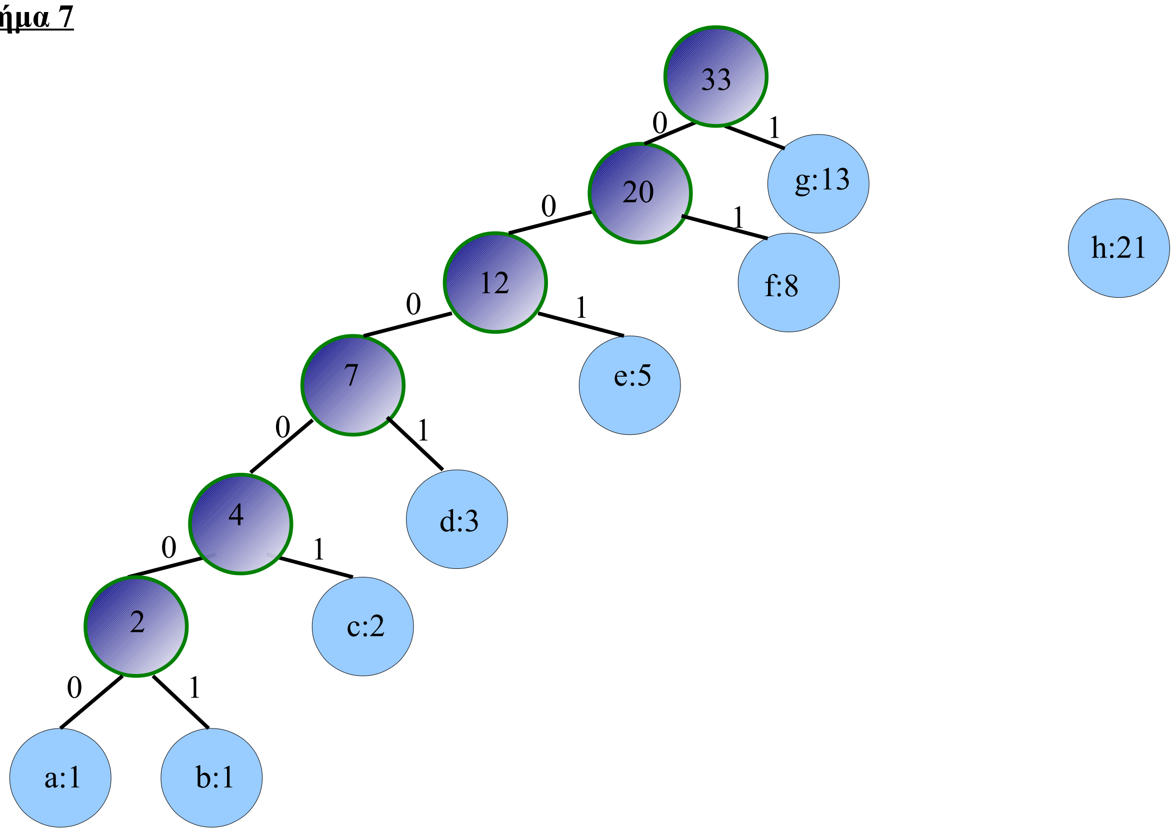
Βήμα 5



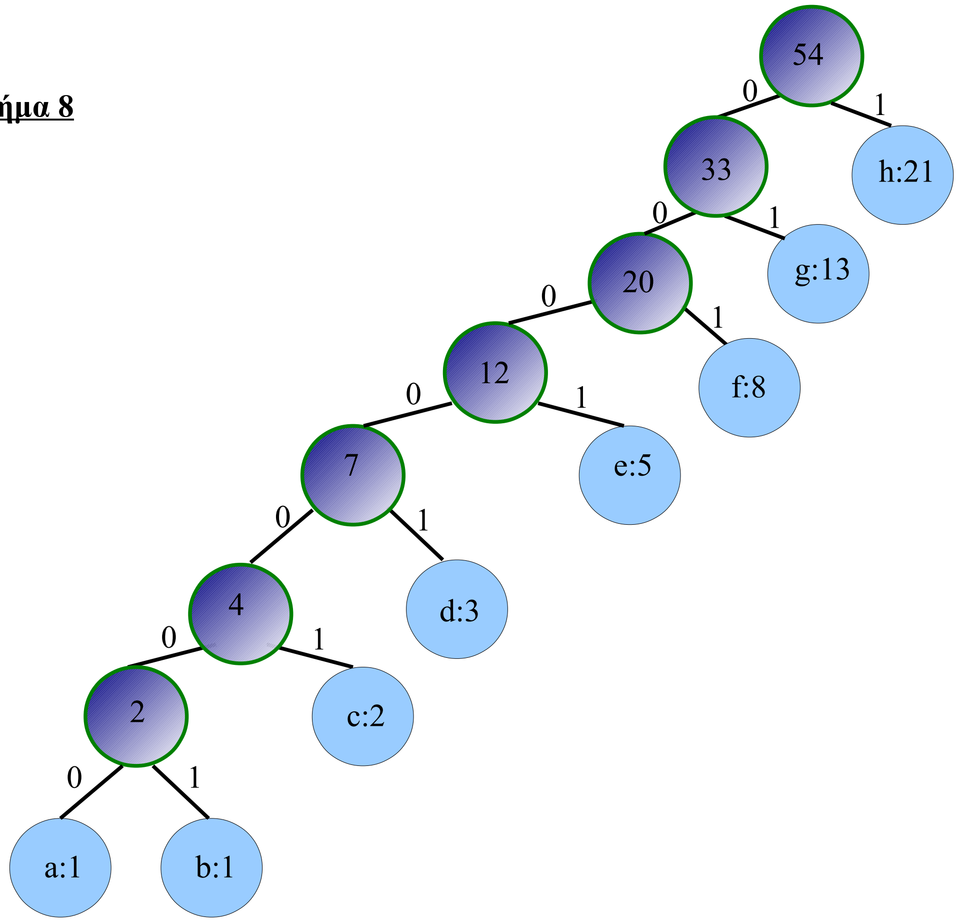
Βήμα 6


20
0
1
12
0
1
f:8
g:13
h:21
7
0
1
e:5
4
0
1
d:3
2
0
1
c:2
a:1
b:1

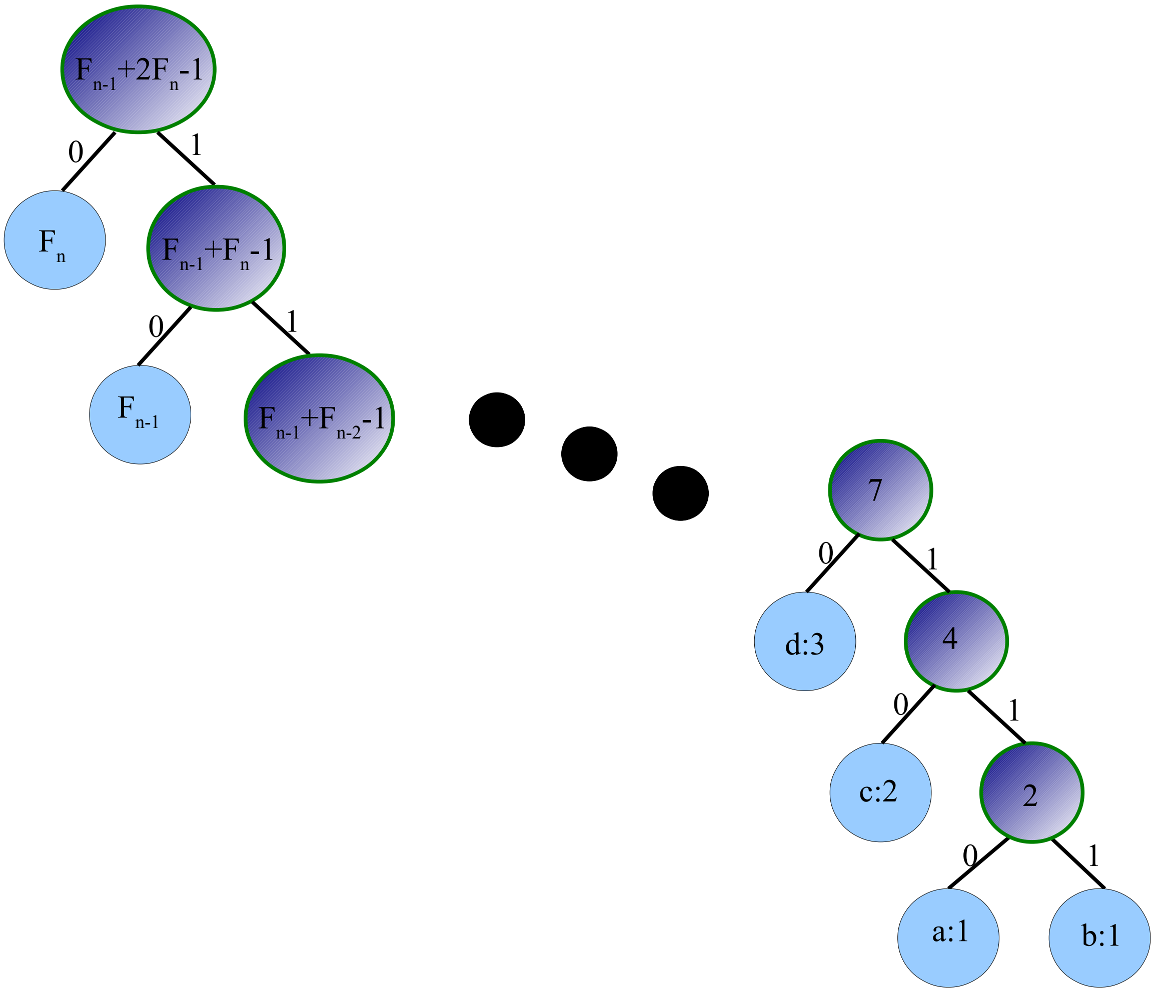
Βήμα 7


33
0
1
20
0
1
g:13
h:21
12
0
1
f:8
7
0
1
e:5
4
0
1
d:3
2
0
1
c:2
a:1
b:1

Βήμα 8



2. Για τους n πρώτους αριθμούς Fibonacci, η κωδικοποίηση Huffman είναι η εξής:



3. Ο αλγόριθμος Huffman με 3 ψηφία κωδικοποίησης είναι ο εξής:
- Καταγραφή συμβόλων και των αντίστοιχων πιθανοτήτων.
- Δημιουργείται ένας κόμβος για κάθε σύμβολο στον οποίο σημειώνεται η αντίστοιχη πιθανότητα.
- Ακολουθεί εύρεση των τριών μικρότερων κόμβων οι οποίοι δεν έχουν κόμβο-πατέρα, και στη συνέχεια δημιουργείται ένας νέος διακλαδιζόμενος κόμβος στον οποίο σημειώνεται το άθροισμα των πιθανοτήτων που έχουν οι τρεις κόμβοι-παιδιά.
- Επαναλαμβάνεται το 3ο βήμα μέχρι όλοι οι κόμβοι εκτός από τη ρίζα να έχουν κόμβο-πατέρα.
- Σημειώνεται με 0, 1 και 2 κάθε τριάδα ακμών (0 για το αριστερό παιδί, 1 για το μεσαίο και 2 για το δεξί).

Ο αλγόριθμος είναι ιδιαίτερα χρήσιμος στη πληροφορική, σε περιπτώσεις που απαιτείται συμπίεση δεδομένων όπως π.χ. Στις JPEG εικόνες και στα αρχεία ήχου (MP3).

4.

n = 5, α = {8, 12, 15, 16, 13}, c = {27, 9, 30, 16, 6} και b = 45

1. Συνεχής έκδοση με $x_i \in [0,1]$

- Αρχικά ταξινομούμε τους λόγους $\frac{c_i}{a_i}$ ως προς φθίνουσα σειρά.

$$\frac{c_1}{a_1} = \frac{27}{8} = 3{,}375\ ,\ \frac{c_2}{a_2} = \frac{9}{12} = 0{,}75\ ,\ \frac{c_3}{a_3} = \frac{30}{15} = 2\ ,\ \frac{c_4}{a_4} = \frac{16}{16} = 1\ ,\ \frac{c_5}{a_5} = \frac{6}{13} \approx 0{,}46$$

Άρα θα έχουμε

$$\frac{c_1}{a_1} = \frac{27}{8} = 3{,}375\ ,\ \frac{c_2}{a_2} = \frac{30}{15} = 2\ ,\ \frac{c_3}{a_3} = \frac{16}{16} = 1\ ,\ \frac{c_4}{a_4} = \frac{9}{12} = 0{,}75\ ,\ \frac{c_5}{a_5} = \frac{6}{13} = 0{,}46$$

- Για όλα τα αντικείμενα θέτουμε $x_i = 0$ (υποθέτουμε δηλαδή ότι δε θα επιλέξουμε κανένα αντικείμενο)
- i=1, z=0 (ξεκίνα με το πρώτο ταξινομημένο και θέσε το συνολικό κέρδος ίσο με μηδέν)
- Όσο $b \geq α_i$ επανέλαβε (Όσο μπορούμε να δεχτούμε κι άλλο αντικείμενο)

  $x_i = 1$ (Θα πάρουμε ολόκληρο το αντικείμενο)

  $b = b - α_i$

  (Το διαθέσιμο βάρος που μπορούμε να προσθέσουμε μειώνεται αφού προσθέσαμε το τρέχων αντικείμενο)

  $z = z + c_i$ (Το συνολικό κέρδος αυξάνεται)

  $i = i + 1$ (Επόμενο αντικείμενο)
- $x_i = b/a_i$ (Το αντικείμενο δε χωράει ολόκληρο, οπότε παίρνουμε ένα μέρος του)
- $z = z + c_i * x_i$ (Το συνολικό κέρδος αυξάνεται)

Επανάληψη	**α**	**b**	**x**	**c**	**z**
1	8	~~45~~,37	1	27	27
2	15	~~37~~,22	1	30	57
3	16	~~22~~,6	1	16	43
-	12	6	$\frac{6}{12} = \frac{1}{2}$	9	$43 + 9 \cdot \frac{1}{2} = 47{,}5$

2. Διακριτή έκδοση. Έστω το σύνολο Υ που θα περιέχει τα αντικείμενα που θα επιλεχθούν.

- Ταξινομούμε τους λόγους $\frac{c_i}{a_i}$ ως προς φθίνουσα σειρά. (Έχει ήδη γίνει στη συνεχή έκδοση)
- Το σύνολο Y είναι αρχικά το κενό σύνολο
- Για όλα τα αντικείμενα
  Αν το b είναι μεγαλύτερο ή ίσο του βάρους του τρέχοντος αντικειμένου
    Πρόσθεσε στο σύνολο Y το τρέχων αντικείμενο
    Μείωσε τη τιμή του b κατά το βάρος του τρέχοντος αντικειμένου
- Το σύνολο Y περιέχει όλα τα αντικείμενα που επιλέχθηκαν

Επανάληψη	**α**	**b**	**Y**
1	8	~~45~~, 37	x_1
2	15	~~37~~,22	x_1 , x_3
3	16	~~22~~,6	x_1 , x_3 , x_4
4	12	6	x_1 , x_3 , x_4
5	13	6	**x_1 , x_3 , x_4**

Άρα θα επιλέξουμε το 1°, το 3° και το 4° αντικείμενο.

3. n = 5, α = {8, 12, 15, 16, 13}, c = {27, 9, 30, 16, 6} και b = 15

Εφαρμόζουμε το πρώτο κομμάτι του αλγορίθμου.

y	**$α_1$**	**$f_1(y)$**	**y**	**$α_1$**	**$f_1(y)$**	**y**	**$α_1$**	**$f_1(y)$**
0	8	0	6	8	0	12	8	27
1	8	0	7	8	0	13	8	27
2	8	0	8	8	27	14	8	27
3	8	0	9	8	27	15	8	27
4	8	0	10	8	27			
5	8	0	11	8	27			

Ο πίνακας των λύσεων έως τώρα

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)

Εφαρμόζουμε το δεύτερο κομμάτι του αλγορίθμου

k	**y**	**$α_{k+1}$**	**$f_k(y)$**	**$f_{k+1}(y)$**	**c_{k+1}**	**x^y_{k+1}**
1	0	12	0	0	-	0
1	1	12	0	0	-	0
1	2	12	0	0	-	0
1	3	12	0	0	-	0
1	4	12	0	0	-	0
1	5	12	0	0	-	0
1	6	12	0	0	-	0
1	7	12	0	0	-	0
1	8	12	27	27	-	0
1	9	12	27	27	-	0
1	10	12	27	27	-	0

k	**y**	**α_{k+1}**	**$f_k(y)$**	**$f_{k+1}(y)$**	**c_{k+1}**	**x^y_{k+1}**
1	11	12	27	27	-	0
1	12	12	27	27	9	0
1	13	12	27	27	9	0
1	14	12	27	27	9	0
1	15	12	27	27	9	0
2	0	15	0	0	-	0
2	1	15	0	0	-	0
2	2	15	0	0	-	0
2	3	15	0	0	-	0
2	4	15	0	0	-	0
2	5	15	0	0	-	0
2	6	15	0	0	-	0
2	7	15	0	0	-	0
2	8	15	27	27	-	0
2	9	15	27	27	-	0
2	10	15	27	27	-	0
2	11	15	27	27	-	0
2	12	15	27	27	-	0
2	13	15	27	27	-	0
2	14	15	27	27	-	0
2	15	15	27	30	30	1
3	0	16	0	0	-	0
3	1	16	0	0	-	0
3	2	16	0	0	-	0
3	3	16	0	0	-	0
3	4	16	0	0	-	0
3	5	16	0	0	-	0
3	6	16	0	0	-	0
3	7	16	0	0	-	0
3	8	16	27	27	-	0
3	9	16	27	27	-	0
3	10	16	27	27	-	0
3	11	16	27	27	-	0
3	12	16	27	27	-	0
3	13	16	27	27	-	0
3	14	16	27	27	-	0
3	15	16	30	30	-	0

k	y	α_{k+1}	$f_k(y)$	$f_{k+1}(y)$	c_{k+1}	x^y_{k+1}
4	0	13	0	0	-	0
4	1	13	0	0	-	0
4	2	13	0	0	-	0
4	3	13	0	0	-	0
4	4	13	0	0	-	0
4	5	13	0	0	-	0
4	6	13	0	0	-	0
4	7	13	0	0	-	0
4	8	13	27	27	-	0
4	9	13	27	27	-	0
4	10	13	27	27	-	0
4	11	13	27	27	-	0
4	12	13	27	27	-	0
4	13	13	27	27	6	0
4	14	13	27	27	6	0
4	15	13	30	30	6	0

Επομένως, ο πίνακας των αποτελεσμάτων είναι ο εξής

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)	27(1)
2	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)
3	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	**30(1)**
4	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	30(0)
5	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	27(0)	30(0)

Εφαρμόζοντας τον αλγόριθμο κατασκευής βέλτιστης λύσης βρίσκουμε ότι θα επιλέξουμε το 3^o αντικείμενο.

5.

Διάμετρος ενός δέντρου είναι η μέγιστη απόσταση μεταξύ δύο κόμβων σε ένα δέντρο. Σύμφωνα με το τελευταίο, μπορούμε να δούμε ότι η μέγιστη απόσταση μεταξύ δύο κόμβων θα είναι πάντα μεταξύ των κόμβων φύλλων. Για να βρούμε λοιπόν την απόσταση ανάμεσα σε δύο κόμβους φύλλα, δηλαδή τη διάμετρο του δέντρου, ακολουθούμε εξής βήματα:

- Κάνουμε μια BFS (κατά πλάτος αναζήτηση) ξεκινώντας από έναν οποιοδήποτε κόμβο φύλλο για να βρούμε έναν κόμβο που να έχει τη μέγιστη απόσταση από τον αρχικό κόμβο που επιλέξαμε. Ο κόμβος που βρήκαμε θα είναι κόμβος φύλλο. Ας το ονομάσουμε κόμβο1.
- Τώρα εφαρμόζουμε και πάλι την BFS ξεκινώντας από τον κόμβο1 και βρίσκουμε τη μέγιστη απόσταση κάθε κόμβου από τον κόμβο1. Η απόσταση αυτή είναι η διάμετρος του δέντρου. Και αυτό επειδή ο κόμβος1 είναι ένα φύλλο του δέντρου και η BFS βρίσκει τη μικρότερη απόσταση από τον κόμβο1 (φύλλο) σε όλους τους άλλους κόμβους στο δέντρο. Σαφώς ο πιο απομακρυσμένος κόμβος από τον κόμβο1 θα είναι ένα φύλλο και ως εκ τούτου θα έχουμε την διάμετρο του δέντρου.

Ο αλγόριθμος καλεί 2 φορές την BFS. Οπότε έχουμε $2\ O(|V|+|E|) = O(|V|+|E|)$

6.

1. Θα ορίσουμε 3 υποπροβλήματα. Η αναδρομική σχέση που συνδέει τη βέλτιστη λύση με τις βέλτιστες λύσεις των υποπροβλημάτων είναι η εξής:
   - f[i, j] το μήκος μιας μέγιστης κοινής υπακολουθίας των ακολουθιών X[i]και Y[j]
   - Αν i = 0 ή j = 0, τότε η μέγιστη κοινή υπακολουθία είναι 0
   - $$f[i,j]=\left\{\begin{matrix} 0,\ \alpha\nu\ i=0\ \acute{\eta}\ j=0 \\ f[i-1,j-1]+1,\ \alpha\nu\ i,j>0\ \kappa\alpha\iota\ x[i]=y[j] \\ max\left(f[i,j-1],f[i-1,j]\right),\ \alpha\nu\, i,j>0\, \kappa\alpha\iota\, x[i]=y[j] \end{matrix}\right\}$$
2. Ο πίνακας λύσεων είναι ο εξής

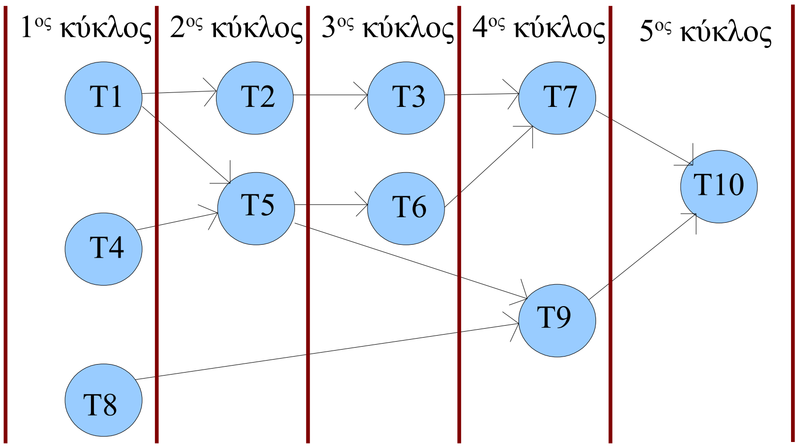
	0 y[j]	1 A	2 L	3 G	4 O	5 R	6 I	7 T	8 H	9 M	10 S
0 x[i]	0	0	0	0	0	0	0	0	0	0	0
1_A	0	D 1	L 1	L 1	L 1	L 1	L 1	L 1	L 1	L 1	L 1
2_L	0	U 1	D 2	L 2	L 2	L 2	L 2	L 2	L 2	L 2	L 2
3_-	0	U 1	U 2	U 2	U 2	U 2	U 2	U 2	U 2	U 2	U 2
4_K	0	U 1	U 2	U 2	U 2	U 2	U 2	U 2	U 2	U 2	U 2
5_H	0	U 1	U 2	U 2	U 2	U 2	U 2	U 2	D 3	L 3	L 3
6_O	0	U 1	U 2	U 2	D 3	L 3	L 3	L 3	U 3	U 3	U 3
7_W	0	U 1	U 2	U 2	U 3	U 3	U 3	U 3	U 3	U 3	U 3
8_A	0	D 1	U 2	U 2	U 3	U 3	U 3	U 3	U 3	U 3	U 3
9_R	0	U 1	U 2	U 2	U 3	D 4	L 4	L 4	L 4	L 4	L 4
10_I	0	U 1	U 2	U 2	U 3	U 4	D 5	U 5	U 5	U 5	U 5
11_Z	0	U 1	U 2	U 2	U 3	U 4	U 5	U 5	U 5	U 5	U 5
12_M	0	U 1	U 2	U 2	U 3	U 4	U 5	U 5	U 5	D 6	L 6
13_I	0	U 1	U 2	U 2	U 2	U 4	D 5	U 5	U 5	U 6	U 6

   Άρα η μέγιστη κοινή υπακολουθία των ακολουθιών “AL-KHOWARIZMI” και “ALGORITHMS” είναι η “¨ALORM”
3. Η πολυπλοκότητα εύρεσης της τιμής της βέλτιστης λύσης είναι O(mn) διότι για να βρούμε την τιμή της βέλτιστης λύσης χρειάζεται να για κάθε χαρακτήρα της ακολουθίας X m χαρακτήρων, κάθε χαρακτήρα της ακολουθίας Y n χαρακτήρων. Άλλωστε για την εύρεση της τιμής αυτής απαιτείται ένας πίνακας m x n.
4. Η πολυπλοκότητα εύρεσης της δομής της βέλτιστης λύσης είναι O(m+n) και αυτό γιατί απλώς χρειάζεται να ξέρουμε το συνολικό μήκος των δύο ακολουθιών. Η ακολουθία X έχει

μήκος m και η ακολουθία Y έχει μήκος n, άρα το συνολικό μήκος είναι m+n.

7.

1. Ο γράφος είναι ο εξής:



2. Ο ελάχιστος αριθμός κύκλων είναι 5. Οι εντολές που εκτελούνται παράλληλα είναι οι εξής:
   - 1ος κύκλος: T1=A*A, T4=B*C, T8=1+C
   - 2ος κύκλος: T2=T1+B, T5=T1+T4, T9=T8+T1
   - 3ος κύκλος: T3=sin(T2), T6=cos(T5)
   - 4ος κύκλος: T7=T3-T6
   - 5ος κύκλος: T10=T7/T9