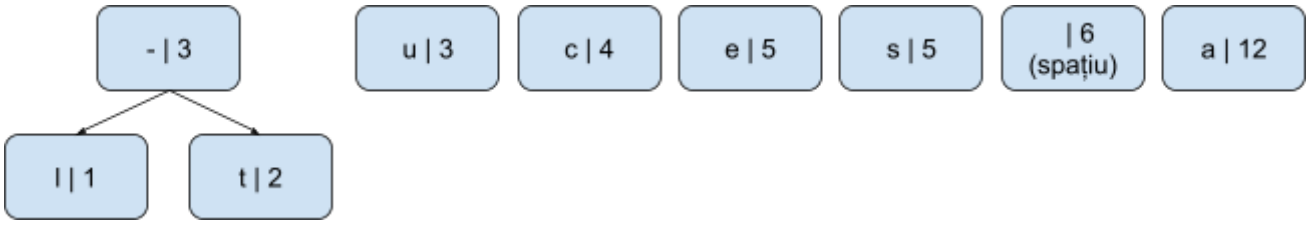
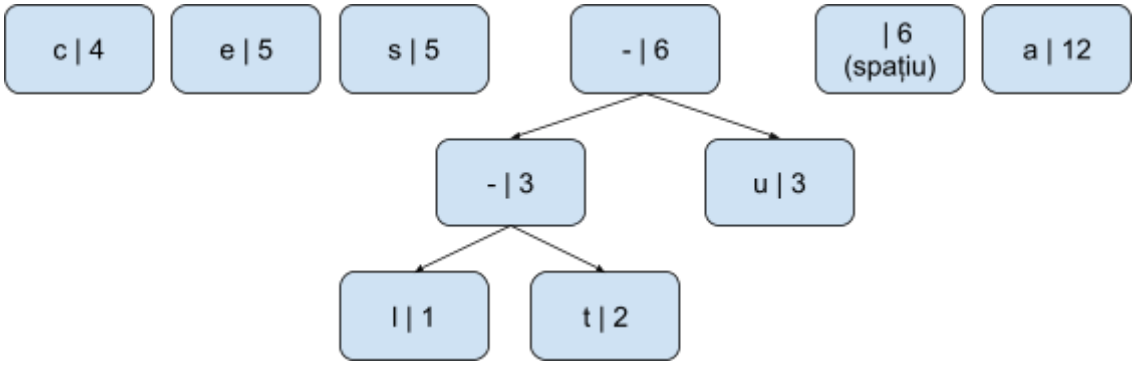
În teoria informației este esențial ca informația, de orice natură ar fi, să fie comprimată atunci când este transmisă pe un canal pentru a optimiza ratele de transfer. O metodă de a efectua această compresie este prin codare. În continuare, trebuie să implementați codarea Huffman pentru un text oarecare. Algoritmul pentru implementare este următorul:

0. Se pornește de la un text sub forma unei secvențe de caractere: “acesta sau aceasta sau acela sau aceea”
1. Se ordonează crescător vectorul de frecvență al caracterelor din text pentru a decide care caractere au o șansă mai mare de apariție. Dacă 2 sau mai multe caractere au aceeași frecvență de apariții, se vor plasa lexicografic în vectorul ordonat:

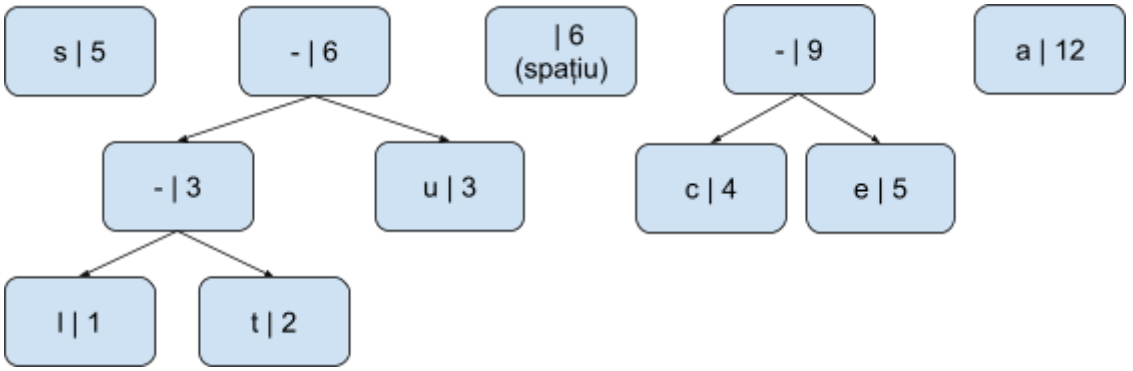
l: 1
t: 2
u: 3
c: 4
e: 5
s: 5
 : 6 (spațiu liber)
a: 12

2. Fiecare dintre caracterele de mai sus va deveni o frunză a arborelui Huffman. Frunzele vor reține caracterul și frecvența sa de apariție, drept dată utilă. Nodurile intermediare ale arborelui vor reține 3 informații:
    - pointer către copilul din stânga;
    - pointer către copilul din dreapta;
    - data utilă = valoare obținută din suma datelor utile ale copiilor săi
3. Pornind de la vectorul de frecvențe de mai sus se parcurg următorii pași:
    - primele 2 valori din vector vor fi eliminate și vor fi utilizate pentru a reprezenta copiii din stânga, respectiv dreapta ai unui nou nod intermediar;
    - noul nod intermediar va fi inserat în vector pe poziție astfel încât să păstreze ordinea (în caz de egalitate, se inserează înaintea elementului de aceeași valoare);
    - se repetă cei 2 pași până când rămâne un singur nod în vector, care va reprezenta rădăcina arborelui.

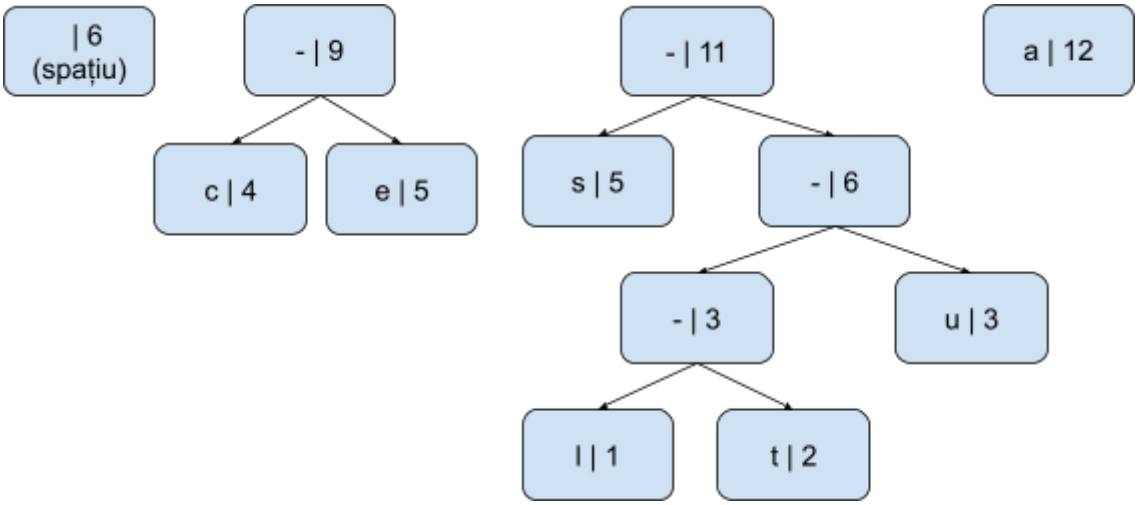
Exemplu:

<table>
<tr><td>3.1.</td><td> 
</td></tr>
<tr><td>3.2.</td><td> 
</td></tr>
</table>

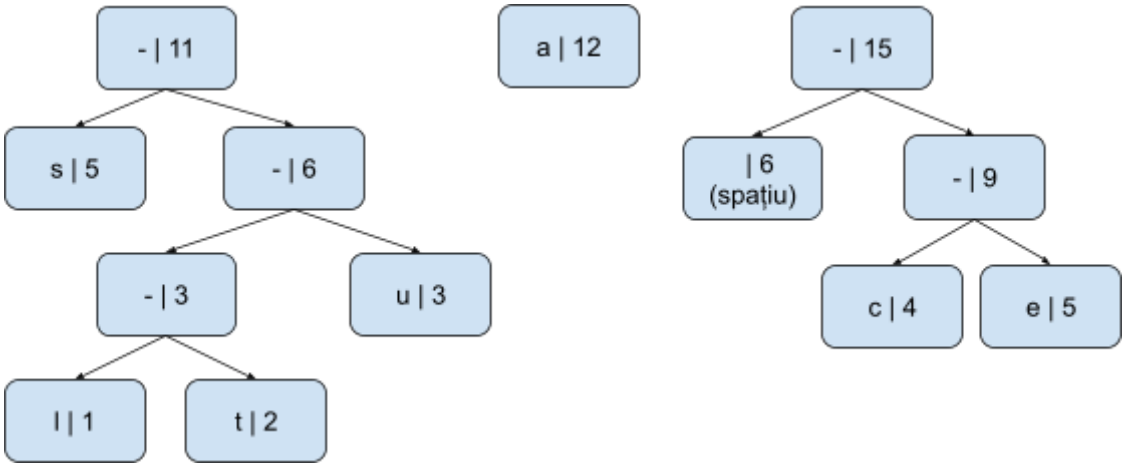
3.3.


s | 5
- | 6
| 6 (spațiu)
- | 9
a | 12
- | 3
u | 3
c | 4
e | 5
l | 1
t | 2

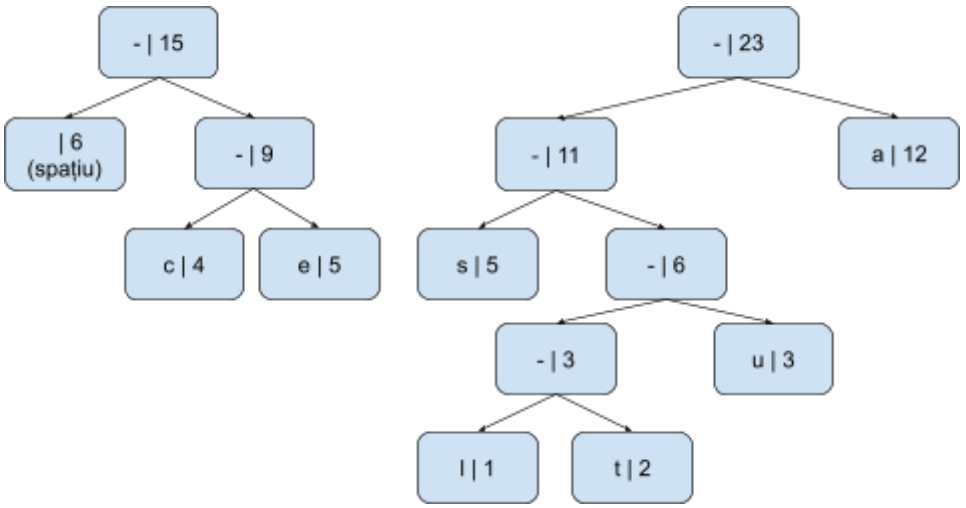
3.4.


| 6 (spațiu)
- | 9
- | 11
a | 12
c | 4
e | 5
s | 5
- | 6
- | 3
u | 3
l | 1
t | 2

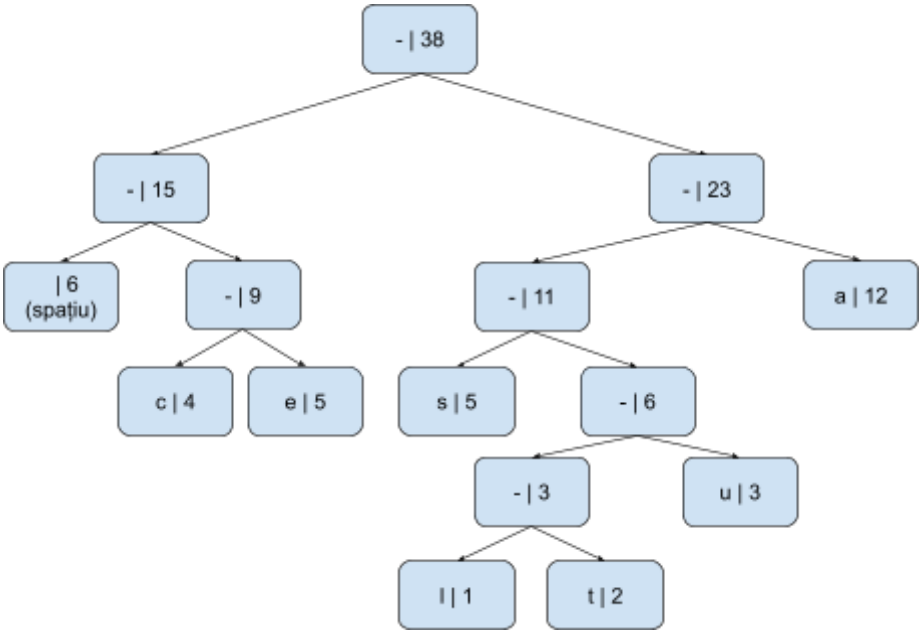
3.5.


- | 11
a | 12
- | 15
s | 5
- | 6
| 6 (spațiu)
- | 9
- | 3
u | 3
c | 4
e | 5
l | 1
t | 2

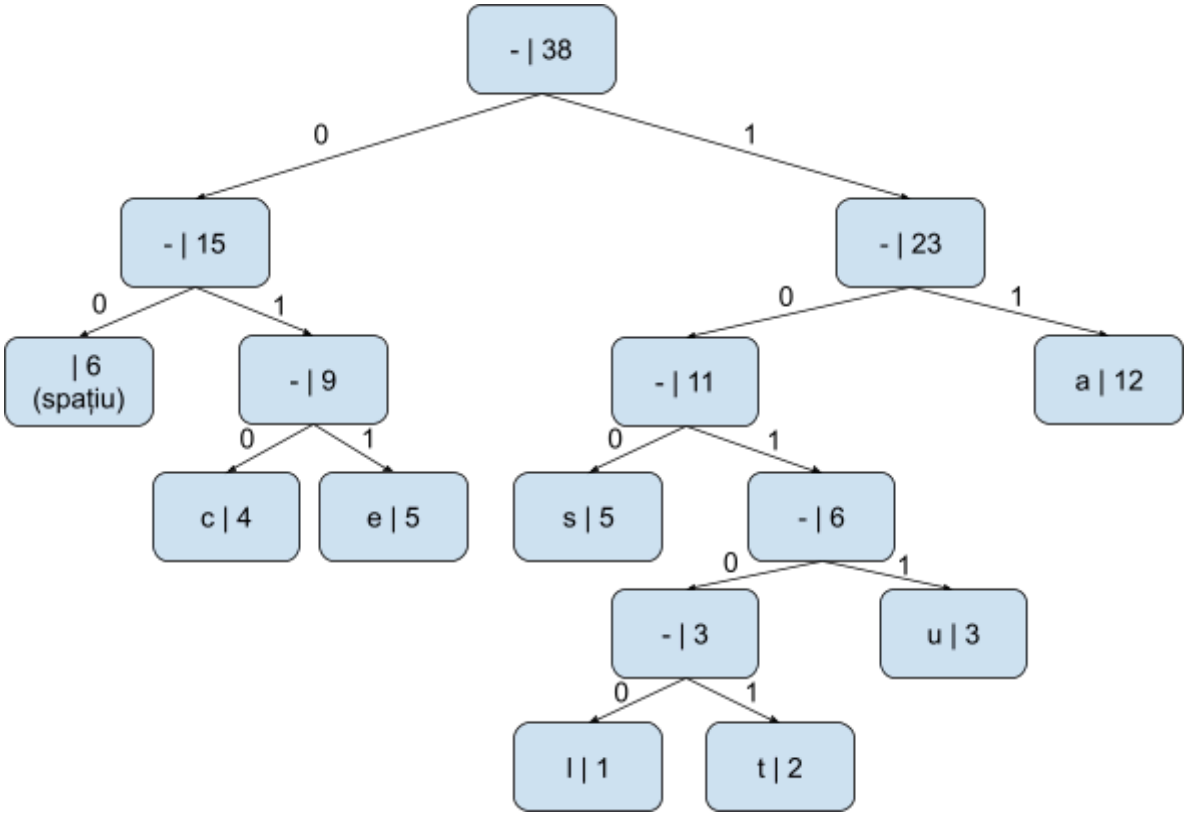
3.6.


- | 15
- | 23
| 6 (spațiu)
- | 9
- | 11
a | 12
c | 4
e | 5
s | 5
- | 6
- | 3
u | 3
l | 1
t | 2

3.7.



4. Fiecărei legături din aborele Huffman îi va fi asociată o valoare binară. Pornind de la rădăcină spre frunze, legăturilor către stânga le va fi asociată valoarea 0, iar celor către dreapta valoarea 1.



5. Se formează codurile binare pentru fiecare caracter prin concatenarea valorilor binare de pe traseul către frunza respectivă.

l: 10100
t: 10101
u: 1011
c: 010
e: 011
s: 100
 : 00 (spațiu liber)
a: 11

Particularitatea codării Huffman constă în faptul că simbolurile mai frecvente au un cod mai scurt pentru a economisi bandă de transfer și niciun simbol nu este prefix pentru vreun alt simbol, pentru a nu crea ambiguități în cod.

6. Fiecare caracter de 8 biți din textul inițial este înlocuit cu codul binar asociat, obținându-se compresia datelor:

acesta sau aceasta sau acela sau aceea

1101001110010101110010011101100110100111110010101110010011101100110100111010011001001110110011010011101001100100111011001101001111

7. Se calculează diferența (în biți) dintre memoria ocupată de textul inițial, unde fiecare caracter ocupă 8 biți și memoria ocupată de secvența binară comprimată.

acesta sau aceasta sau acela sau aceea: 38 caractere * 8b/caracter = 304 biți

1101001110010101110010011101100110100111110010101110010011101100110100111010011001001110110011010011101001100100111011001101001111: 105 biți

Diferența este de 304-105 = 199 biți.

Pentru partea pe care o aveți de rezolvat se dă un text citit de la tastatură până la tasta enter. Ulterior, se citește un întreg cu valoare între 1 și 5, în funcție de care se vor executa următoarele funcționalități:

1. Să se afișeze pe ecran caracterele citite, în ordinea crescătoare a frecvenței de apariții, împreună cu frecvența lor (pasul 1 din exemplu). Dacă 2 litere au aceeași frecvență de apariții se vor ordona alfabetic. Afișarea se va face sub forma:

<caracter>:<valoare frecvență>

2. Să se afișeze pe ecran valorile datelor utile, sortate, după 3 iterații ale algoritmului de generare a arborelui Huffman (valorile de pe prima linie din pasul 3.3 din exemplu).

3. Din arborele Huffman (se dă deja construit) se extrag și se afișează pe ecran caracterele, împreună cu codul asociat, păstrând ordinea caracterelor de la punctul 1 (pasul 5 din exemplu). Se păstrează același format pentru afișare ca la punctul 1.

4. Fiecare caracter din șirul inițial este înlocuit cu codul obținut la punctul 3. Să se afișeze pe ecran codul binar astfel obținut (pasul 6 din exemplu).

5. Să se afișeze pe ecran diferența exprimată în biți dintre memoria ocupată de șirul original și cea ocupată de șirul codat (pasul 7 din exemplu).