Laborator 13 - Cod Huffman

13.2. Coduri Huffman

Codurile Huffman reprezinta o modalitate optima de arhivare a unui text dat. Ideea de baza este ca fiecare caracter va fi inlocuit cu un cod format din biti de 0 si 1 astfel incat un caracter cu frecventa mare de aparitie sa aiba un cod cat mai mic posibil fata de unul cu frecventa mica. Codurile asociate sunt coduri cu prefix (nici un cod valid nu reprezinta prefixul unui alt cod valid). In codarea Huffman textul dat este analizat pentru a se calcula frecventa de aparitie a fiecarui caracter. Se construieste apoi un arbore binar format din noduri frunze (asociate caracterelor) si noduri interne, astfel: se porneste cu n noduri libere (frunze), unde n reprezinta numarul de caractere distincte din text; fiecare nod contine un caracter si frecventa sa de aparitie; se creeaza apoi un nod nou avand ca fii 2 noduri cu frecventa minima de aparitie; se continua analog cu multimea de noduri din care s-au eliminat cele doua noduri folosite si la care s-a adaugat noul nod creat; procesul se incheie atunci cand se ramane cu un singur nod, acesta devenind si radacina unui arbore binar Huffman. Codul unui caracter se calculeaza parcurgand lantul de la radacina la nodul frunza care il contine si considerand 0 daca se merge pe directia fiului stang, respectiv 1 pentru fiul drept. Pentru a se dezarhiva un text codat Huffman trebuie sa se stie care sunt codurile caracterelor (date de arborele Huffman).

Exemplul 13.2. Pentru textul "*this is an example of a huffman tree*" se calculeaza urmatoarele frecvente de aparitie si se creeaza arborele Huffman de mai jos

(vezi http://commons.wikimedia.org/wiki/File:Huffman_ tree.svg)

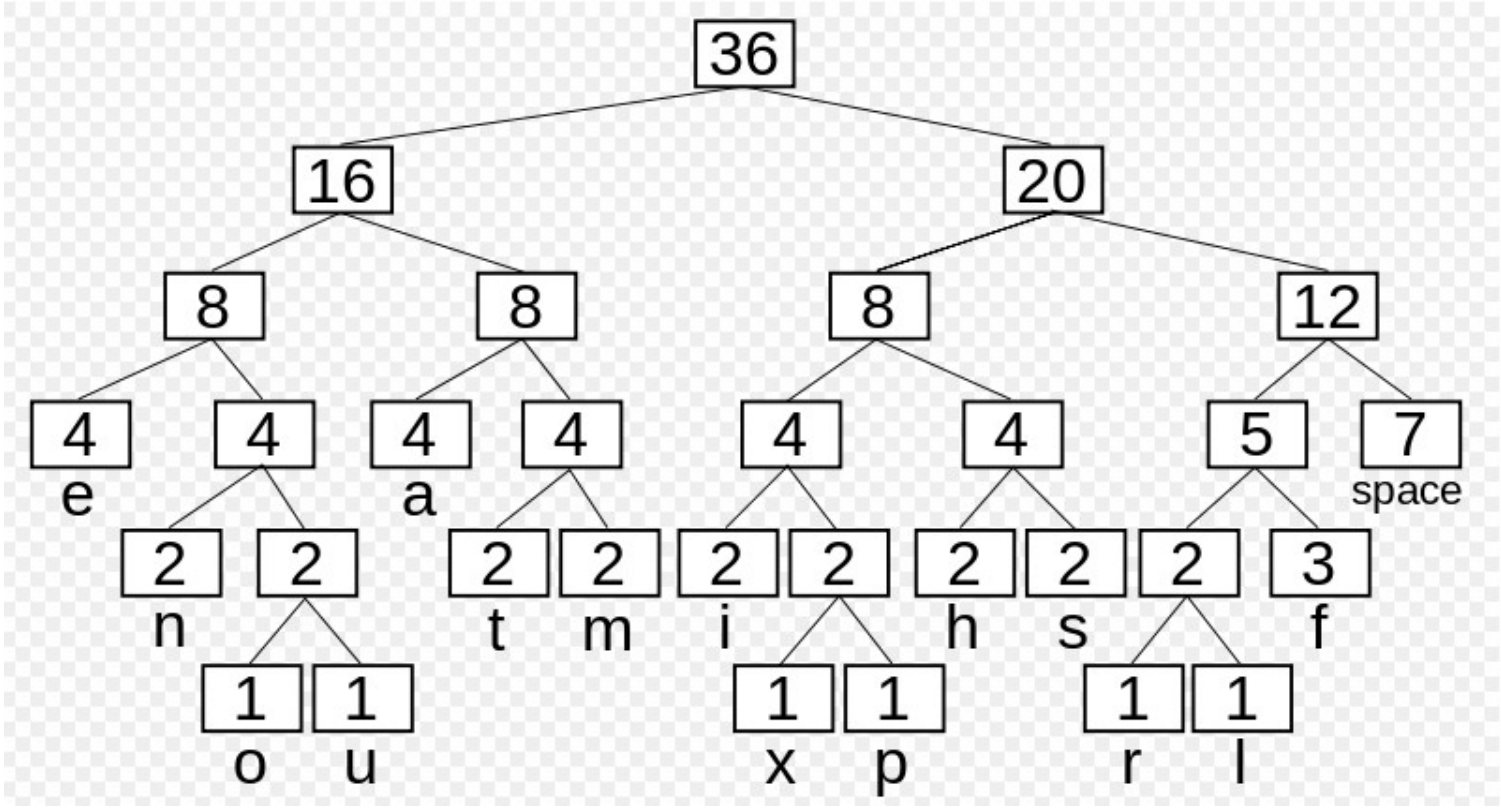
Textul este astfel codat si rezulta arhivarea
"01101010100010111111000101111101000101110001001001001111001111001000111001101101111010111010000111110111010111010001011101101100000000" pe 135 de biti, reprezentand o compresie de 53% fata de numarul de 288 de biti folositi pentru memorarea codurilor ASCII corespunzatoare caracterelor din text (36 de caractere x 8 biti de caracter), parcurgandu-se arborele de la nodul radacina la caracter, astfel: 0 pentru subramura stanga, respectiv 1 pentru dreapta, fara nodul radacina.

Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000

Char	Freq	Code
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110

Char	Freq	Code
p	1	10011
r	1	11000
u	1	00111
x	1	10010

Arborele rezultat este:

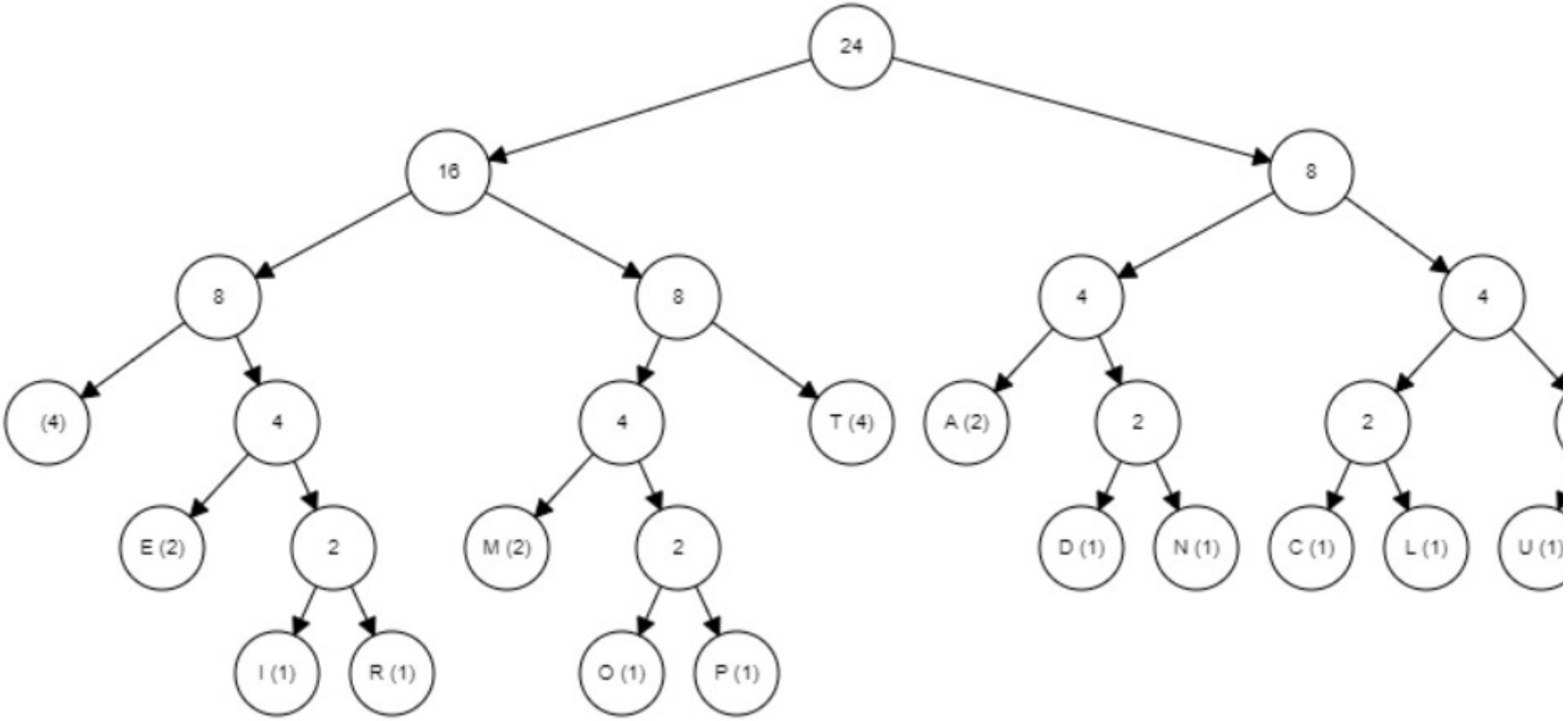


Exercitiul 13.1

Construiti arborele binar asociat codului Huffman pentru textul: *text de comprimat*

Exercitiul 13.2

Pentru arborele Huffman de mai jos:



.

Determinati forma decomprimata a secventei: 1100010100100010110011100110010010011

Exercitiul 13.3 Fisierul date.in contine un text format numai din literele mari ale alfabetului latin.

1. Scrieti un program C prin care caculati frecventa de aparitie a fiecarui caracter din text continut de fisier

Exercitiul 13.4.*. Construiti apoi codul Huffman si arborele asociat acestuia.