

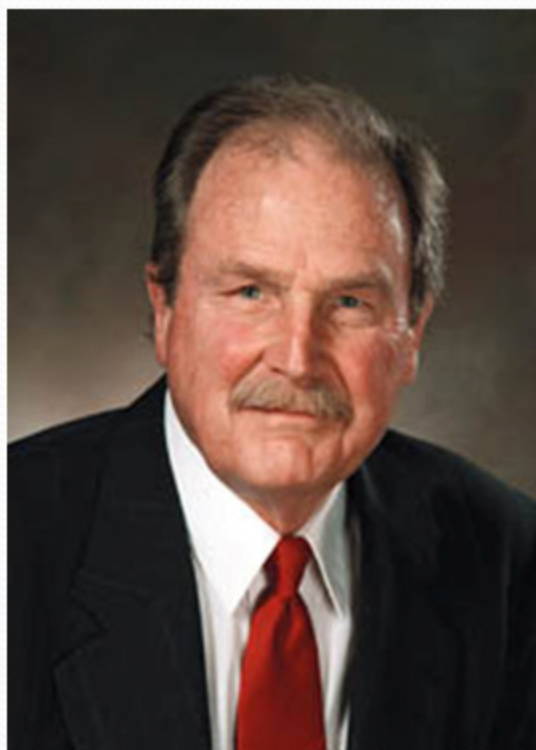
Algorytm kompresji Huffmana

Andrzej Pisarski

Plan seminarium

- David Huffman
- Zastosowania kodowania Huffmana
- Algorytm Huffmana
 - Pojęcia
 - Kompresja
 - Dekompresja

David Huffman



- Ur. 9 sierpnia 1925 - zm. 7 października 1999 w Santa Cruz, Kalifornia.
- Pochodził z Ohio i kształcił się na miejscowym uniwersytecie. Po II wojnie światowej wrócił na uniwersytet stanowy Ohio i uzyskał tam stopień magistra w dziedzinie inżynierii elektrycznej (1949). W 1953 obronił doktorat w Massachusetts Institute of Technology. Pracował później jako wykładowca w MIT, a od 1967 na University of California w Santa Cruz.

http://pl.wikipedia.org/wiki/David_Huffman

David Huffman

- Prowadził badania nad wykończeniem maszyn statycznych, obwodami scalonymi, procedurami syntezy sygnałów i projektami sygnałów.
- Huffman nie patentował swoich osiągnięć, koncentrując się na pracy edukacyjnej. Mawiał: "moimi produktami są studenci".

http://pl.wikipedia.org/wiki/David_Huffman

Zastosowanie kodowania Huffmana

- „Huffman is probably best known for the development of the Huffman Coding Procedure, the result of a term paper he wrote while a graduate student at the Massachusetts Institute of Technology (MIT). "Huffman Codes" are used in nearly every application that involves the compression and transmission of digital data, such as fax machines, modems, computer networks, and high-definition television.”
- Kompresja plików dźwiękowych MP3

<http://www1.ucsc.edu/currents/99-00/10-11/huffman.html>

Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka



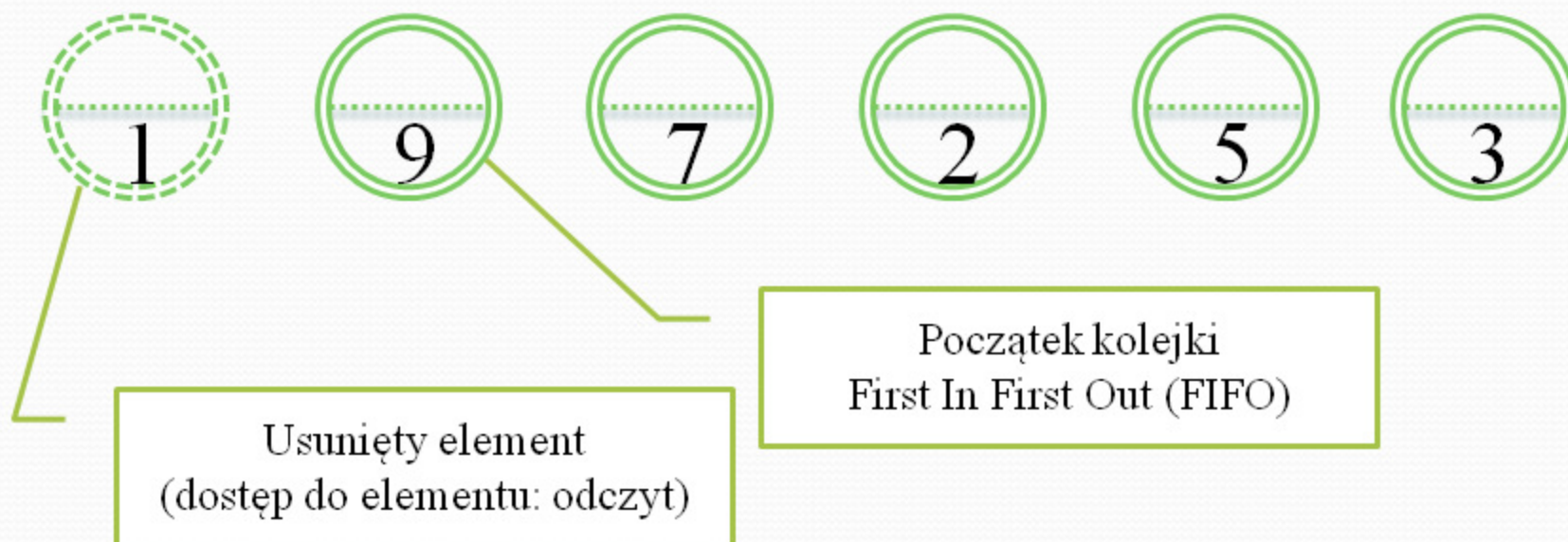
Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka



Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka



Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka priorytetowa



Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka priorytetowa



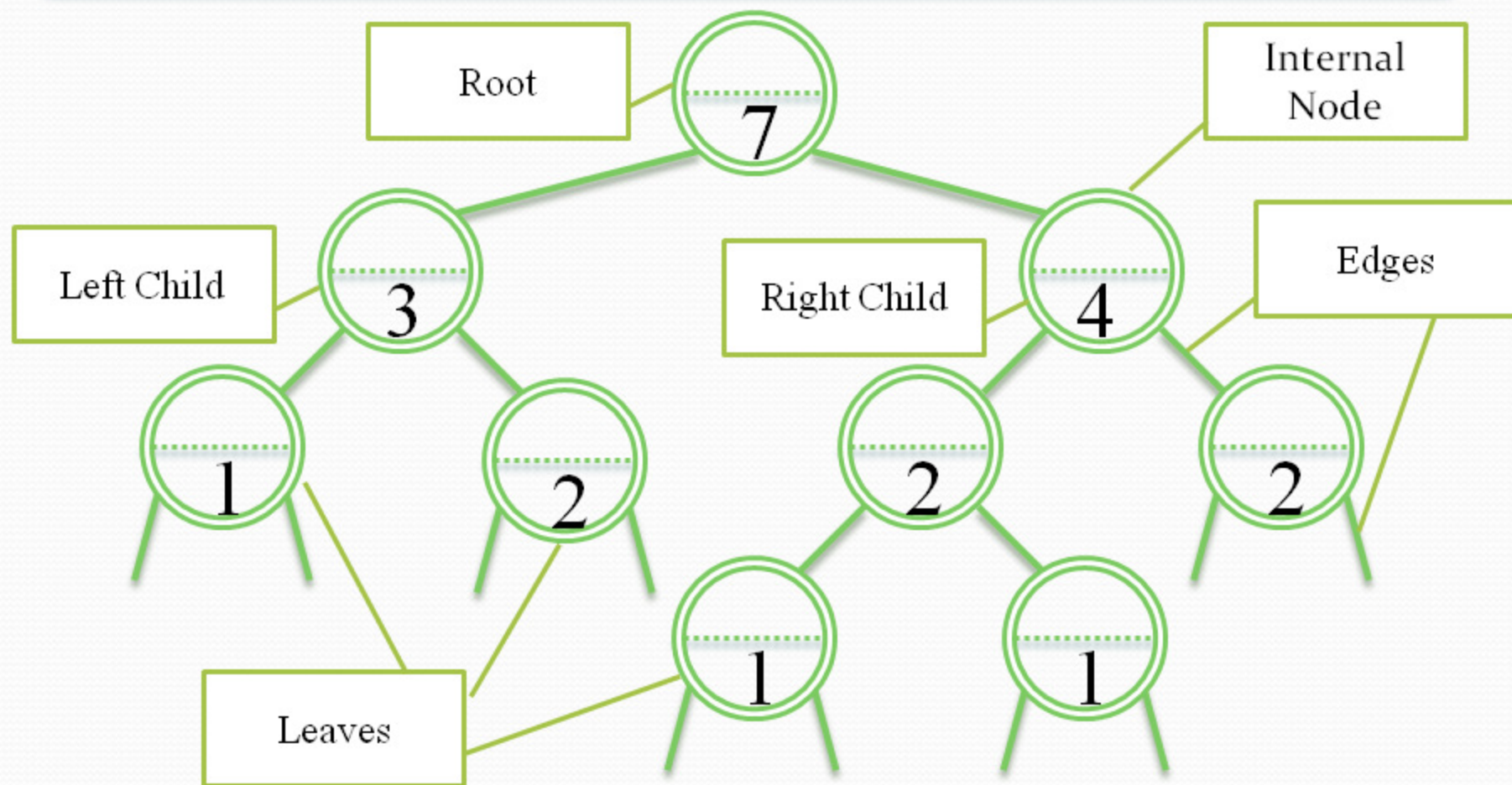
Algorytm Huffmana: pojęcia

Struktura danych typu: kolejka priorytetowa



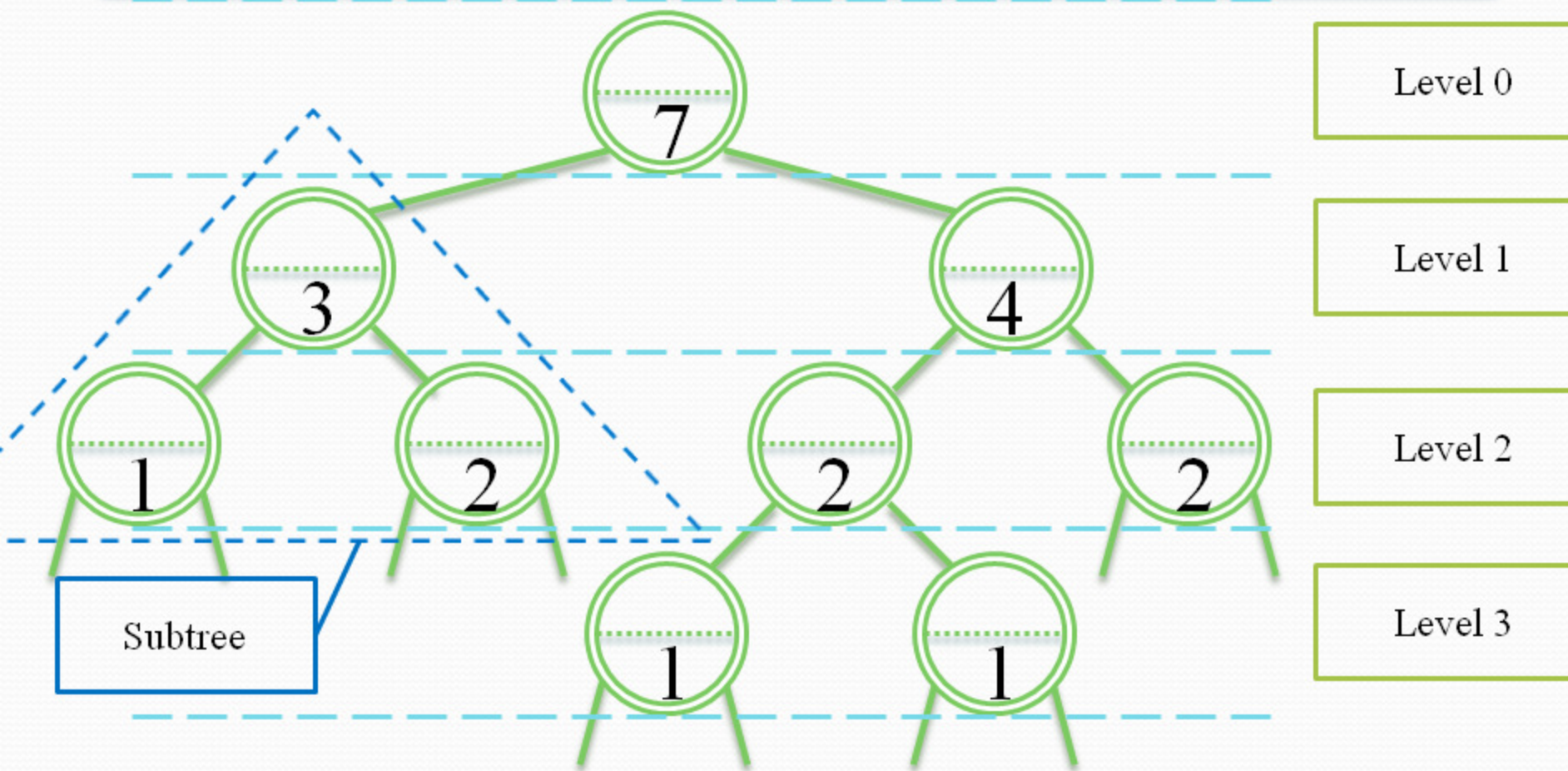
Algorytm Huffmana: pojęcia

Struktura danych typu: drzewo



Algorytm Huffmana: pojęcia

Struktura danych typu: drzewo



Algorytm Huffmana: kompresja

„ABRACADABRA!”

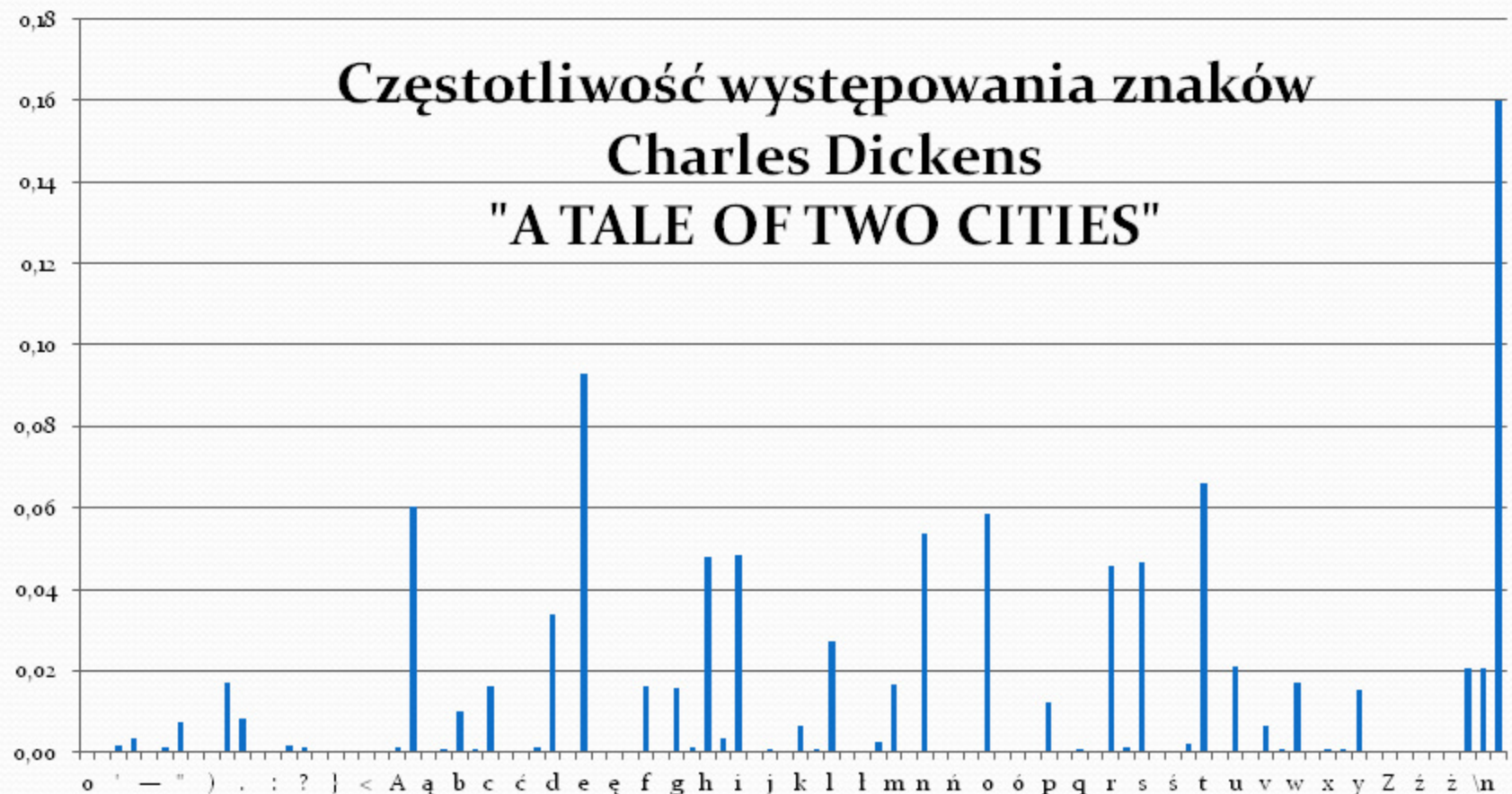
ZNAK	SŁOWA KODOWE		
	Strona kodowa ANSI <u>ASCII</u>	KOD BEZPREFIKSOWY	KOD PREFIKSOWY
!	00100001	1100	100
A	01000001	0	0
B	01000010	101	1
C	01000011	1101	00
D	01000100	100	10
R	01010010	111	11
B [b]	12 [96]	4 [28+4]	3 [18+6]

Algorytm Huffmana: kompresja

1. Wczytać dane wejściowe.
2. Utwórz tablicę częstotliwości wystąpień znaków w danych wejściowych.
3. Utwórz drzewo kodowania Huffmana.
4. Utwórz tablicę słów kodowych.
5. Zapisz drzewo Huffmana (dane wyjściowe).
6. Zapisz liczbę znaków tekstu wejściowego.
7. Zapisz dane wejściowe wykorzystując tablicę słów kodowych.

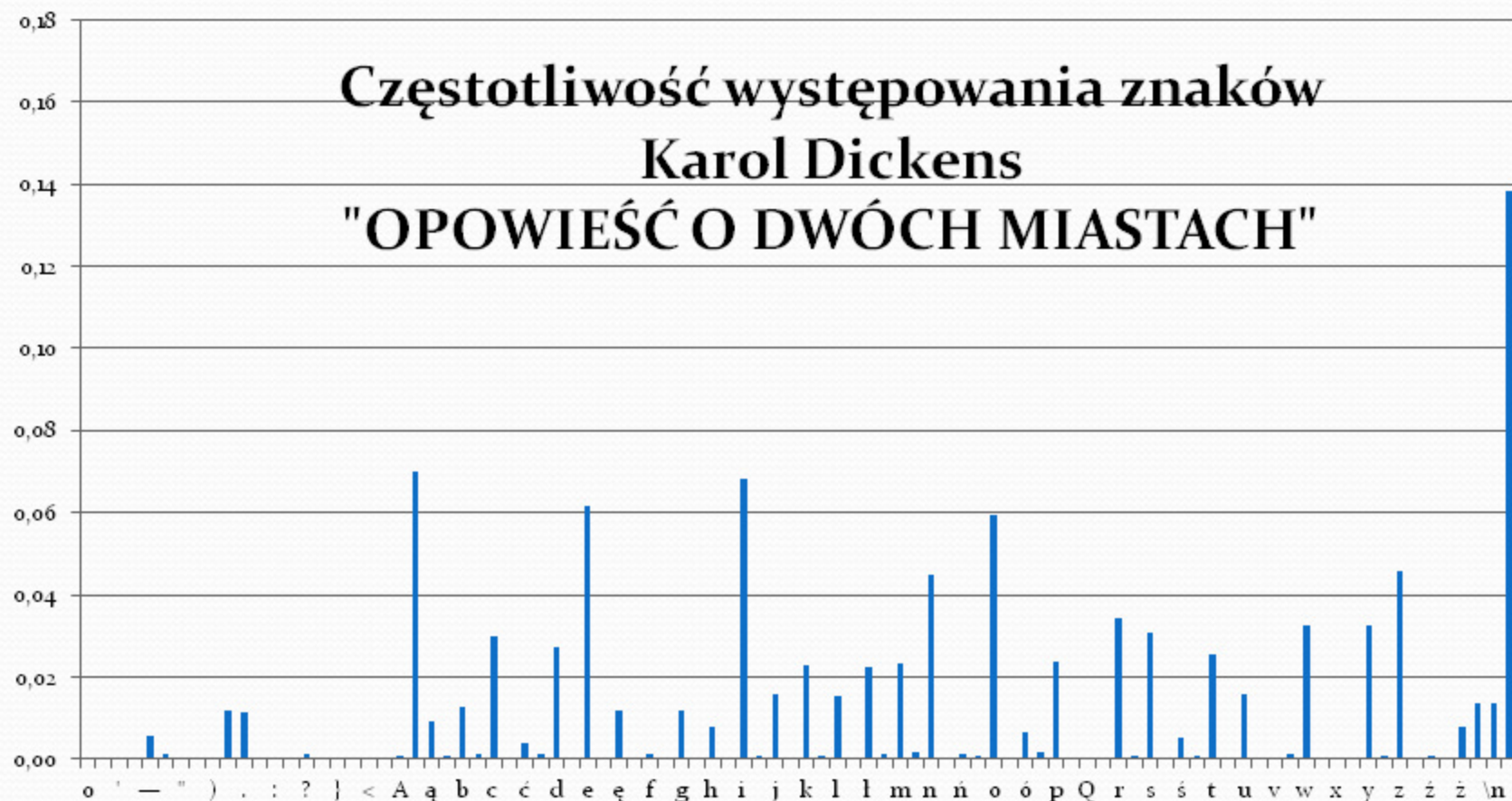
Algorytm Huffmana: kompresja

(2/7) Tablica częstotliwości wystąpień znaków



Algorytm Huffmana: kompresja

(2/7) Tablica częstotliwości wystąpień znaków



Algorytm Huffmana: kompresja

(2/7) Tablica częstotliwości wystąpień znaków

A TALE OF TWO CITIES	
SPACJA	0,1602
e	0,09305
t	0,06591
a	0,06017
o	0,05856
n	0,05356
i	0,04828
h	0,048
s	0,04659
r	0,04562
...	
V	0,00013
X	0,00010
K	0,00008
U	0,00004
Q	0,00004

OPOWIEŚĆ O DWÓCH MIASTACH	
SPACJA	0.1384
a	0.06983
i	0.06841
e	0.06158
o	0.05964
z	0.04577
n	0.04504
r	0.03451
w	0.03275
y	0.03247
...	
q	2.5e-006
}	2.5e-006
/	2.5e-006
Ń	2.5e-006
<	2.5e-006

Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana

„ABRACADABRA!”

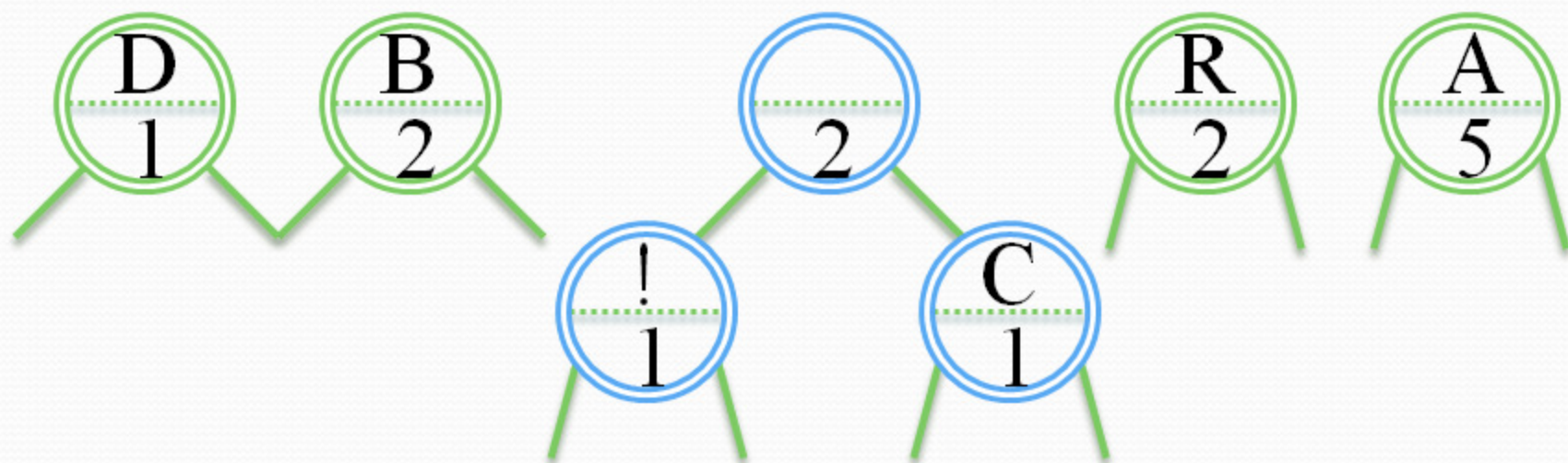


Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana

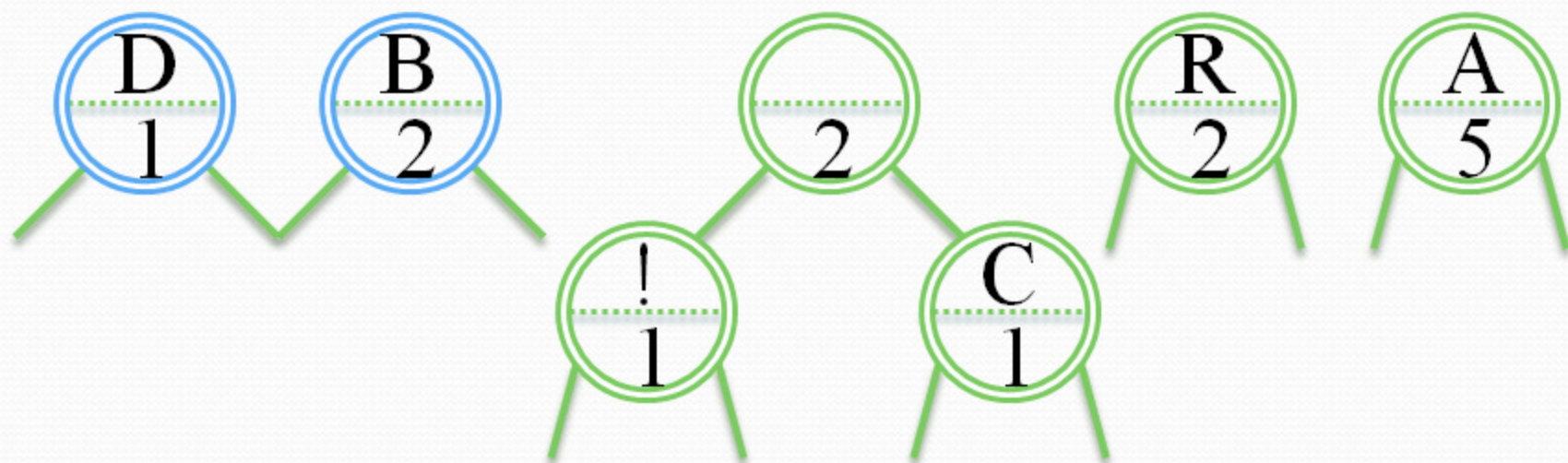
„ABRACADABRA!”



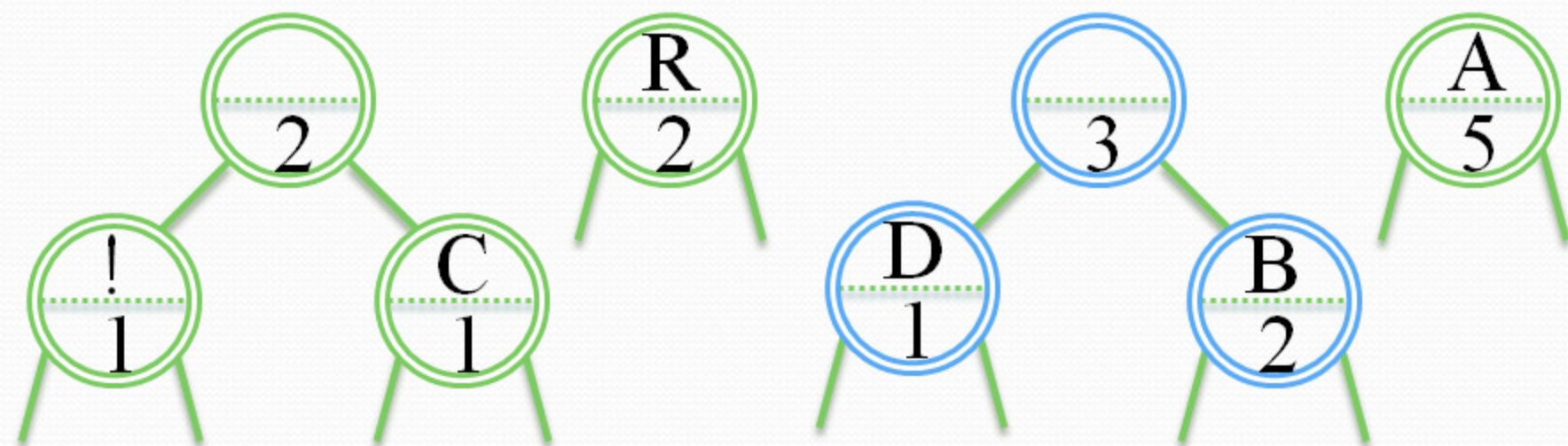
Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana



Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana

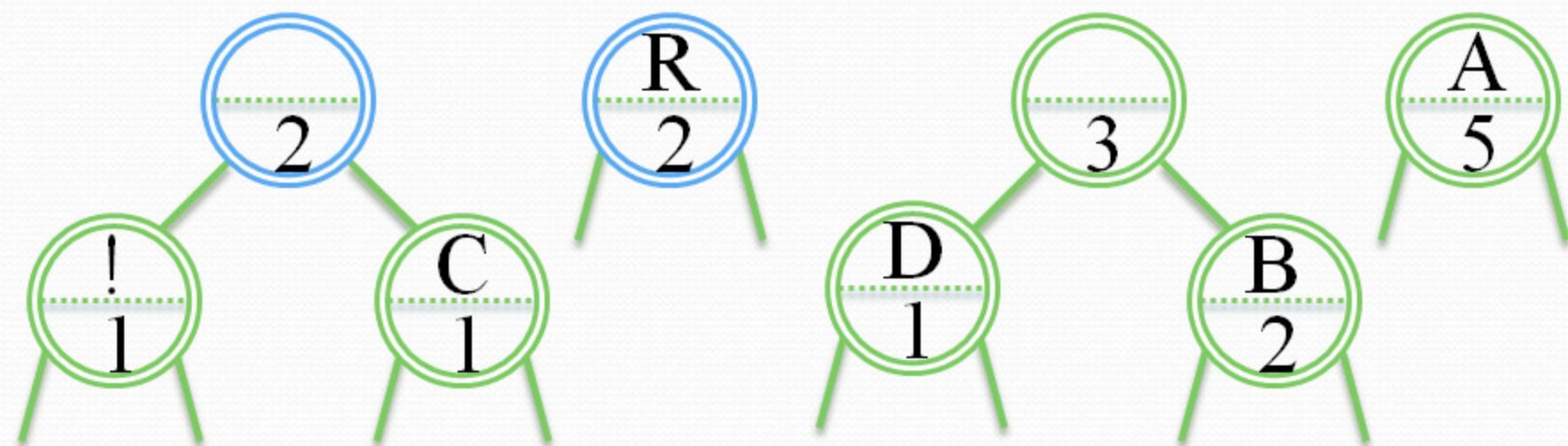


Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana

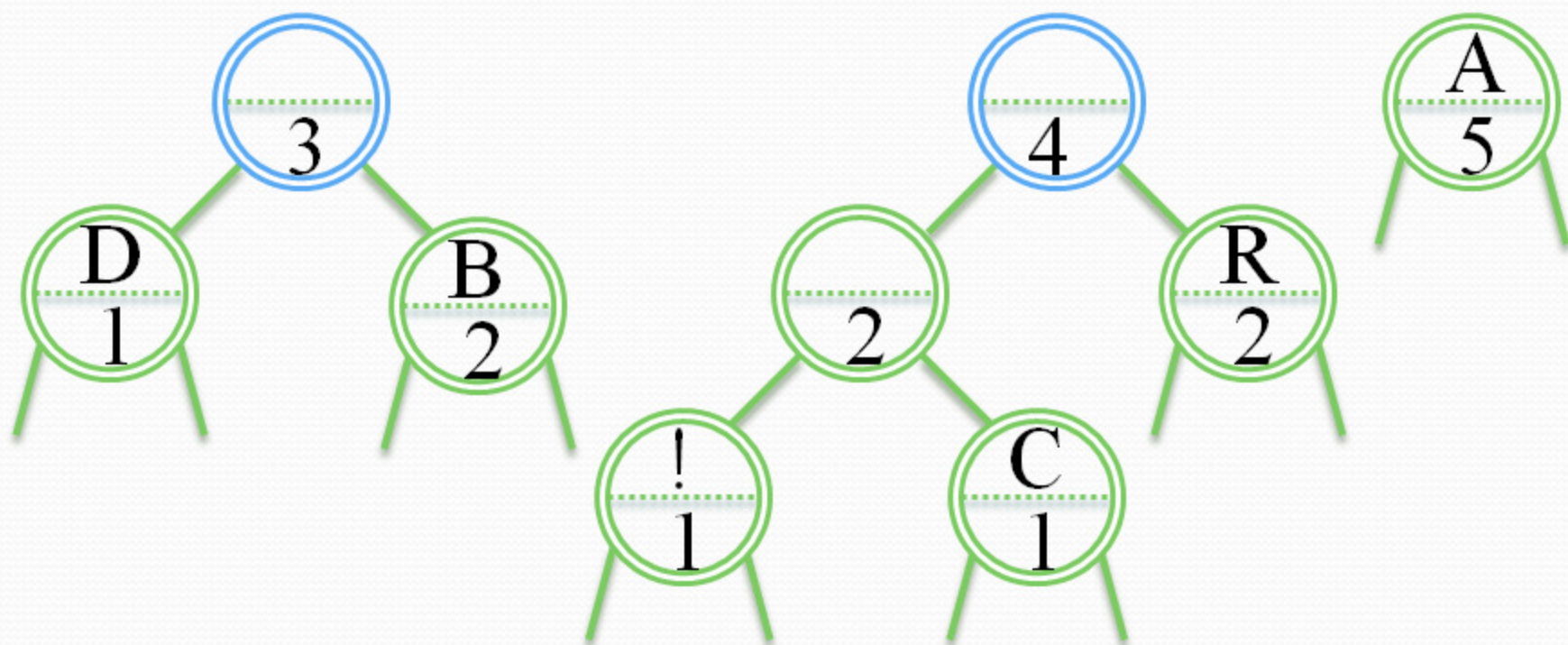


Algorytm Huffmana: kompresja

(3) drzewo kodowania Huffmana

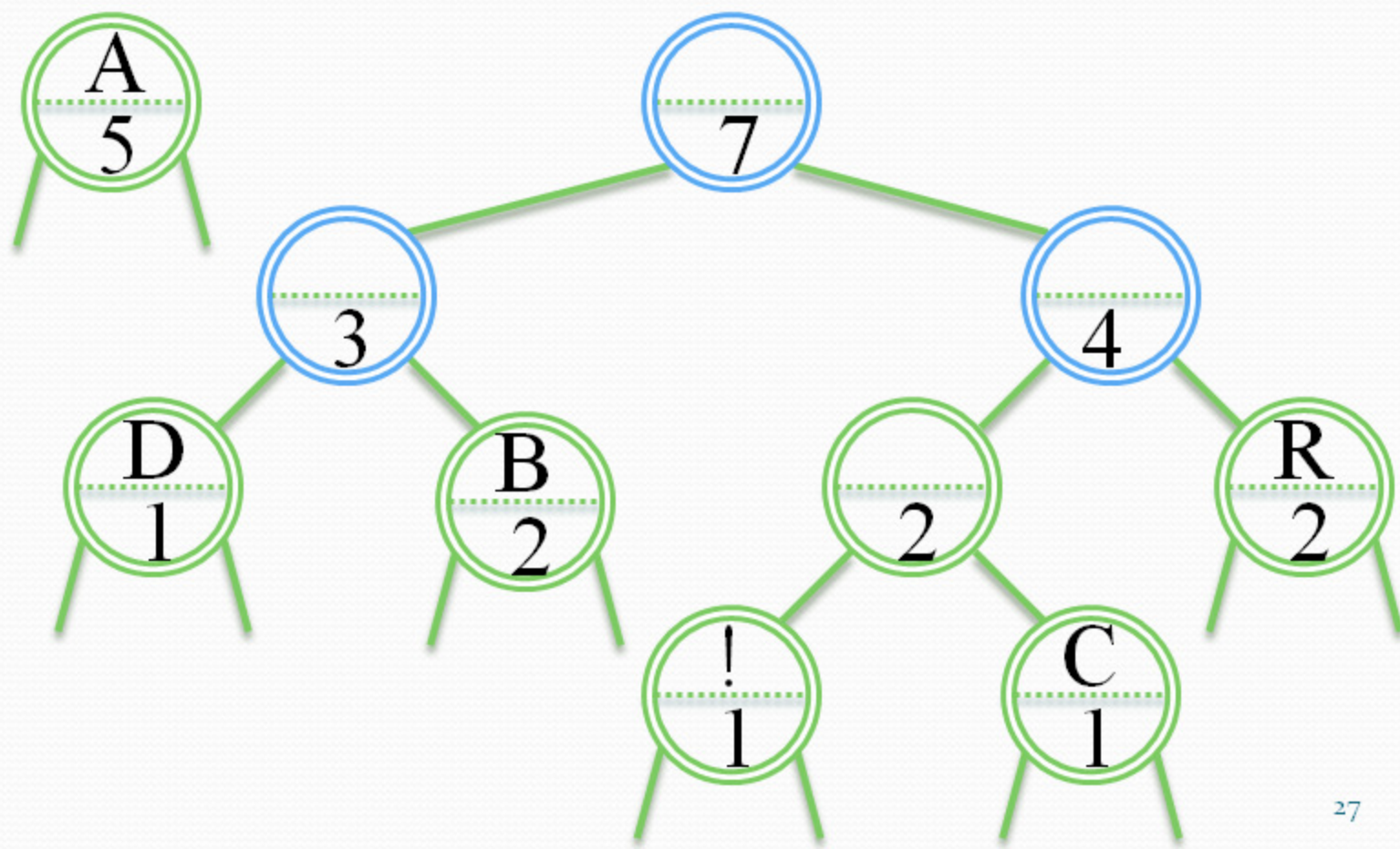


Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana

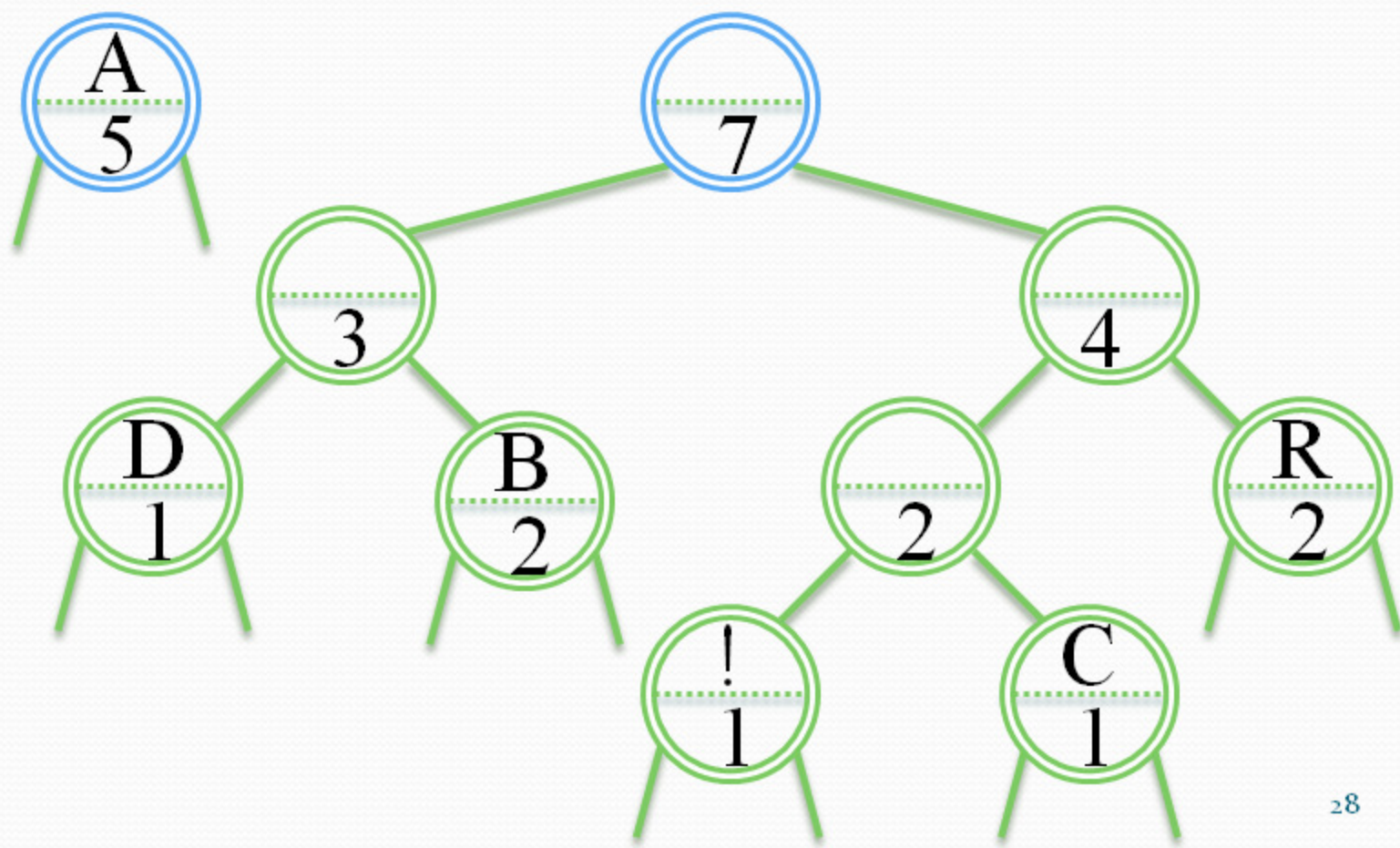


Algorytm Huffmana: kompresja

(3/7) drzewo kodowania Huffmana

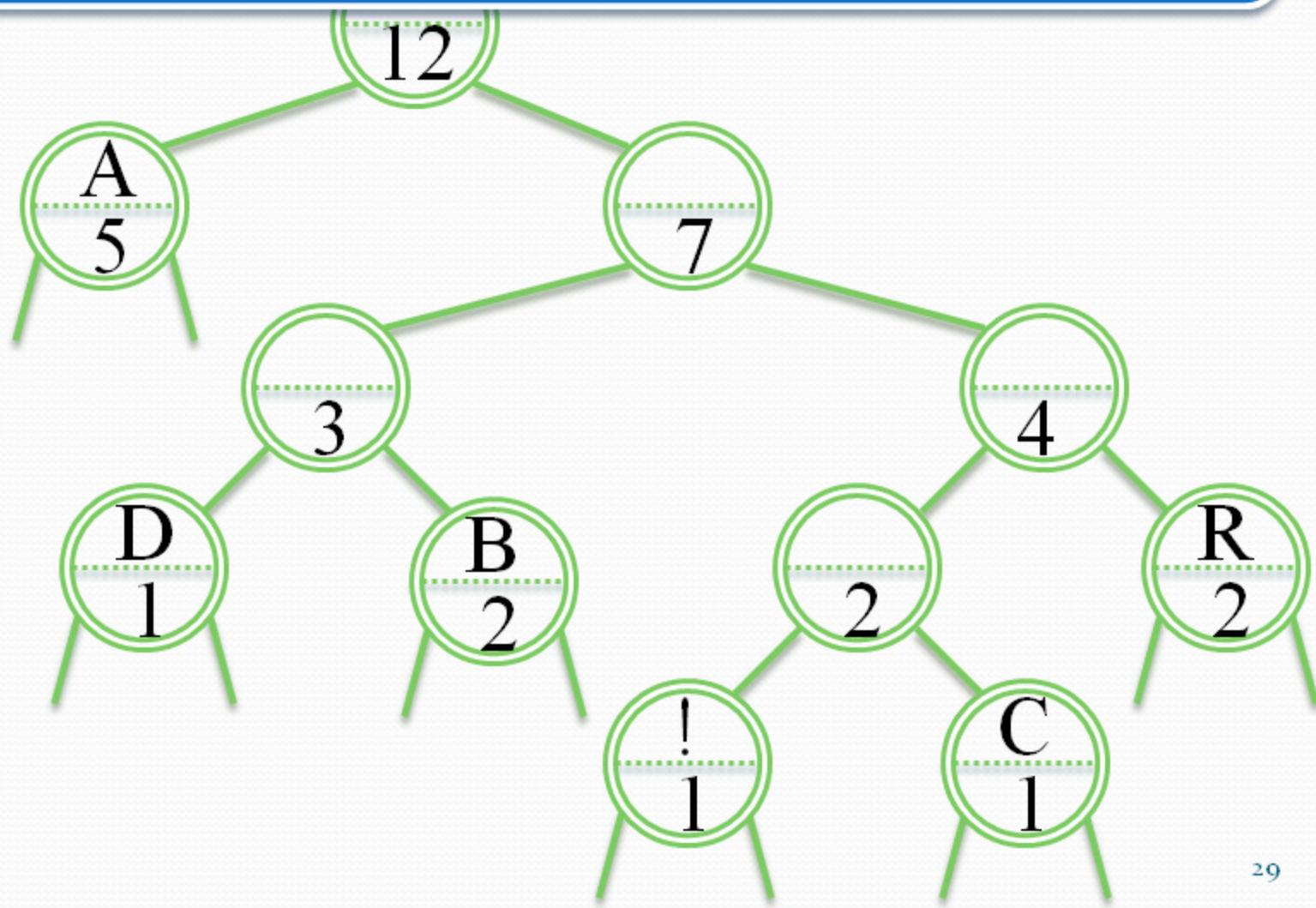


Algorytm Huffmana: kompresja (3/7) drzewo kodowania Huffmana



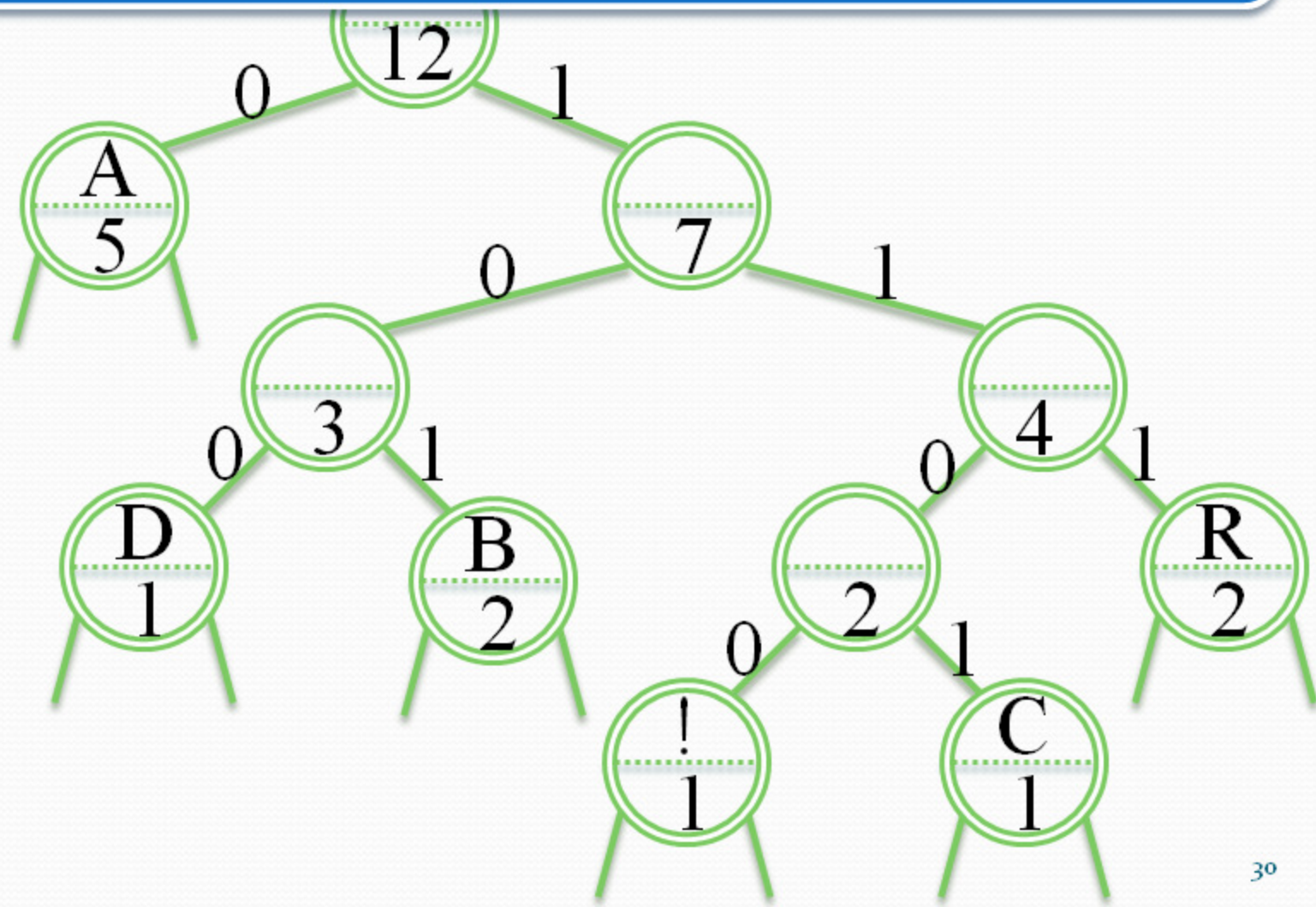
Algorytm Huffmana: kompresja

(3/7) drzewo kodowania Huffmana



Algorytm Huffmana: kompresja

(4/7) tablica słów kodowych



Algorytm Huffmana: kompresja

(4/7) tablica słów kodowych

ZNAK	SŁOWA KODOWE	
	KODOWANIE ANSI	KOD BEZPREFIKSOWY
!	00100001	1100
A	01000001	0
B	01000010	101
C	01000011	1101
D	01000100	100
R	01010010	111

Algorytm Huffmana: kompresja

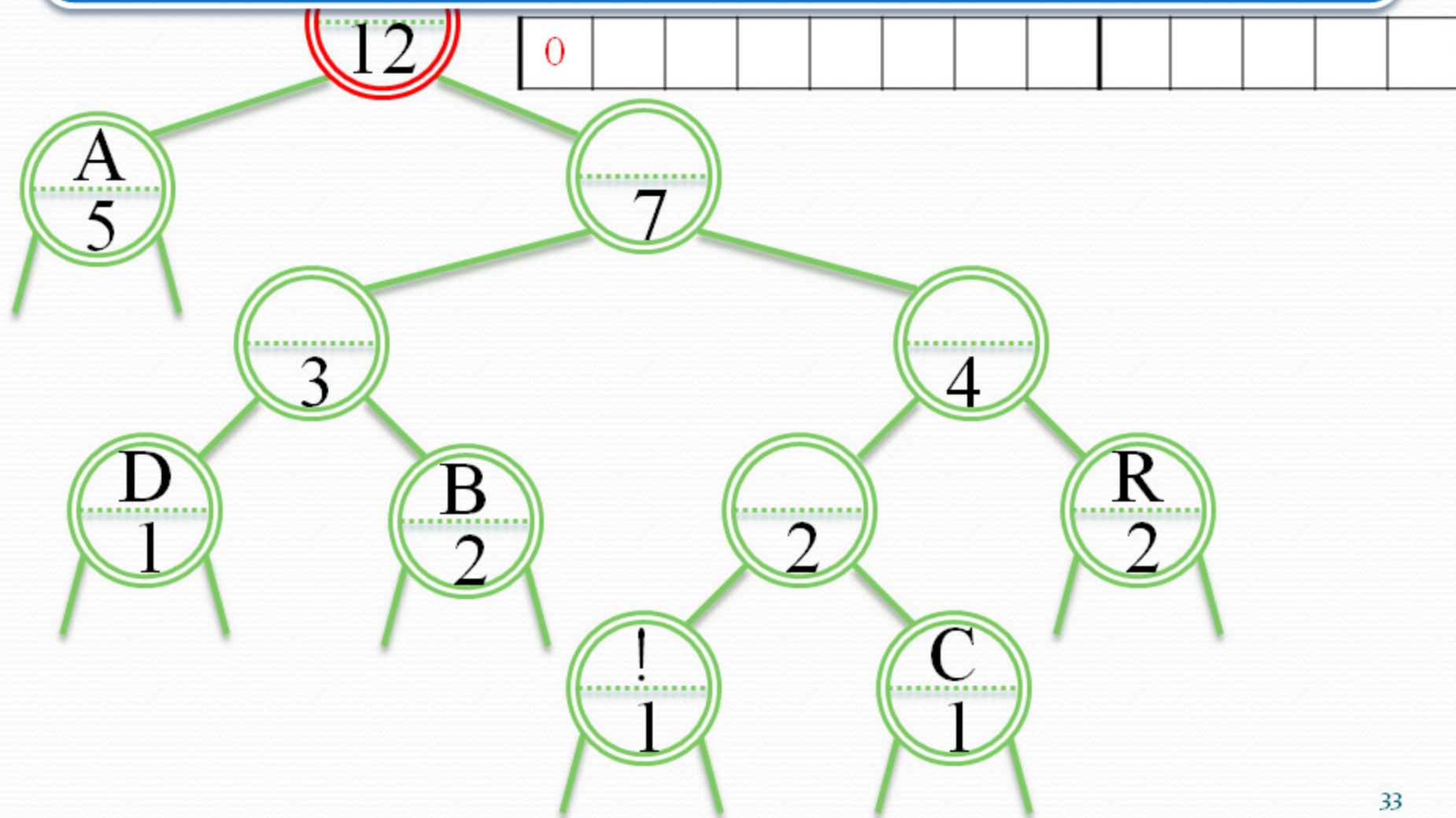
(5/7) zapis drzewa Huffmana (preorder)

```
void writeTree(Node* x)
{
    if(x->isLeaf())
    {
        ///Zapisz:
        '1'          // Lisc
        'znak'       // Znak przechowywany w lisciu
        return;
    }

    ///Zapisz:
    '0'             // Wezel wewnetrzny
    writeTree(x->pLeftChild);
    writeTree(x->pRightChild);
}
```

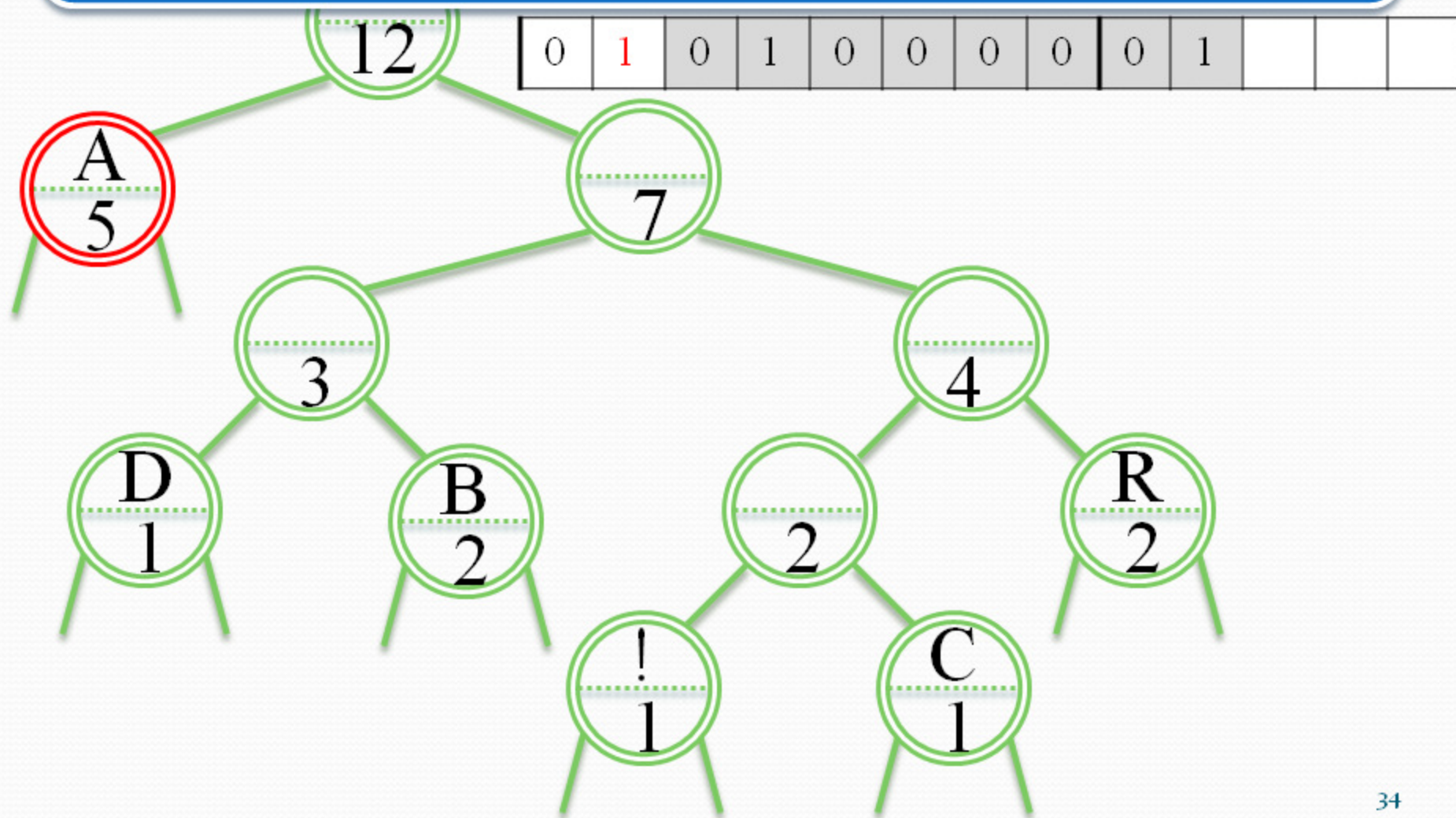
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



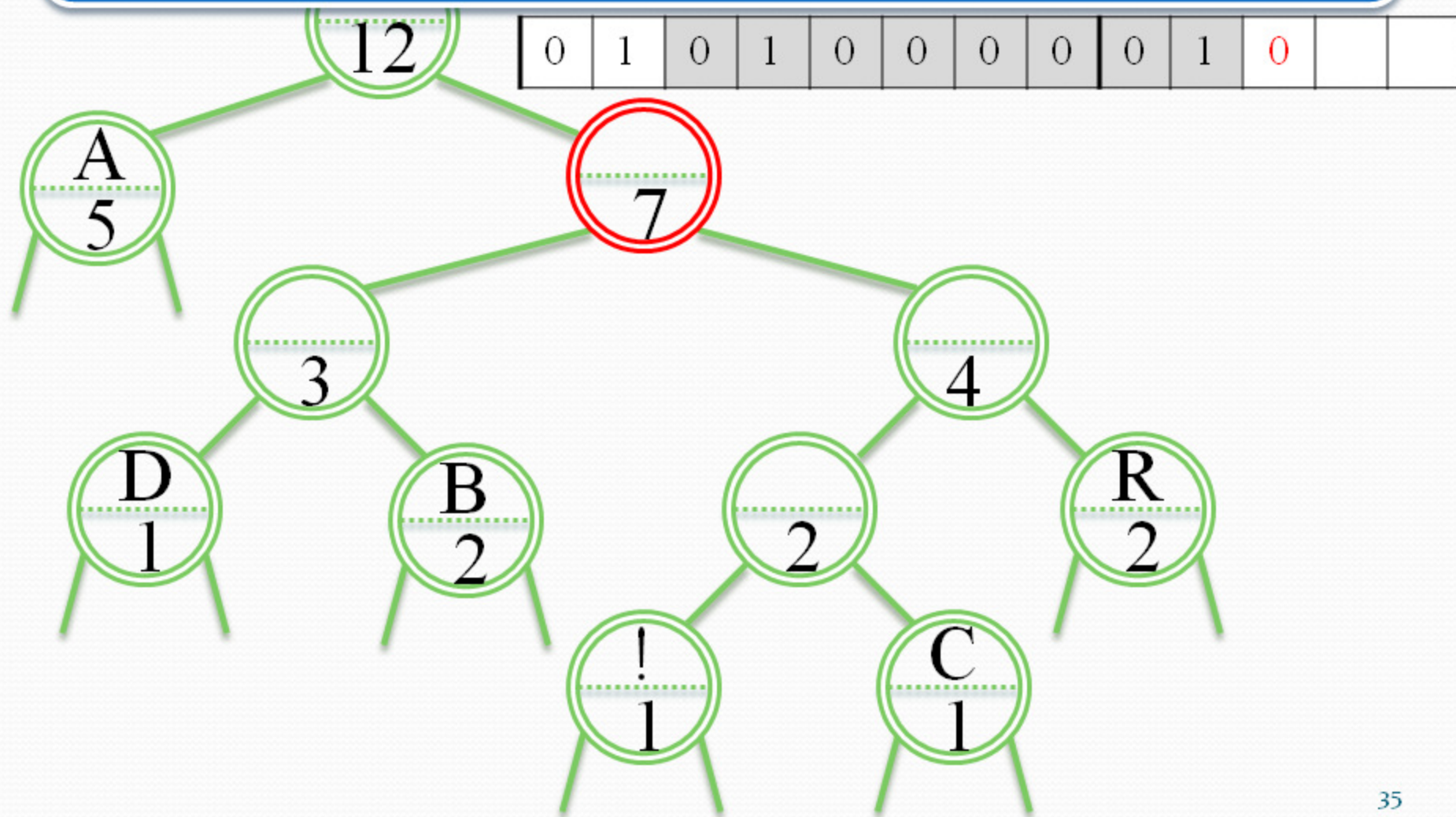
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



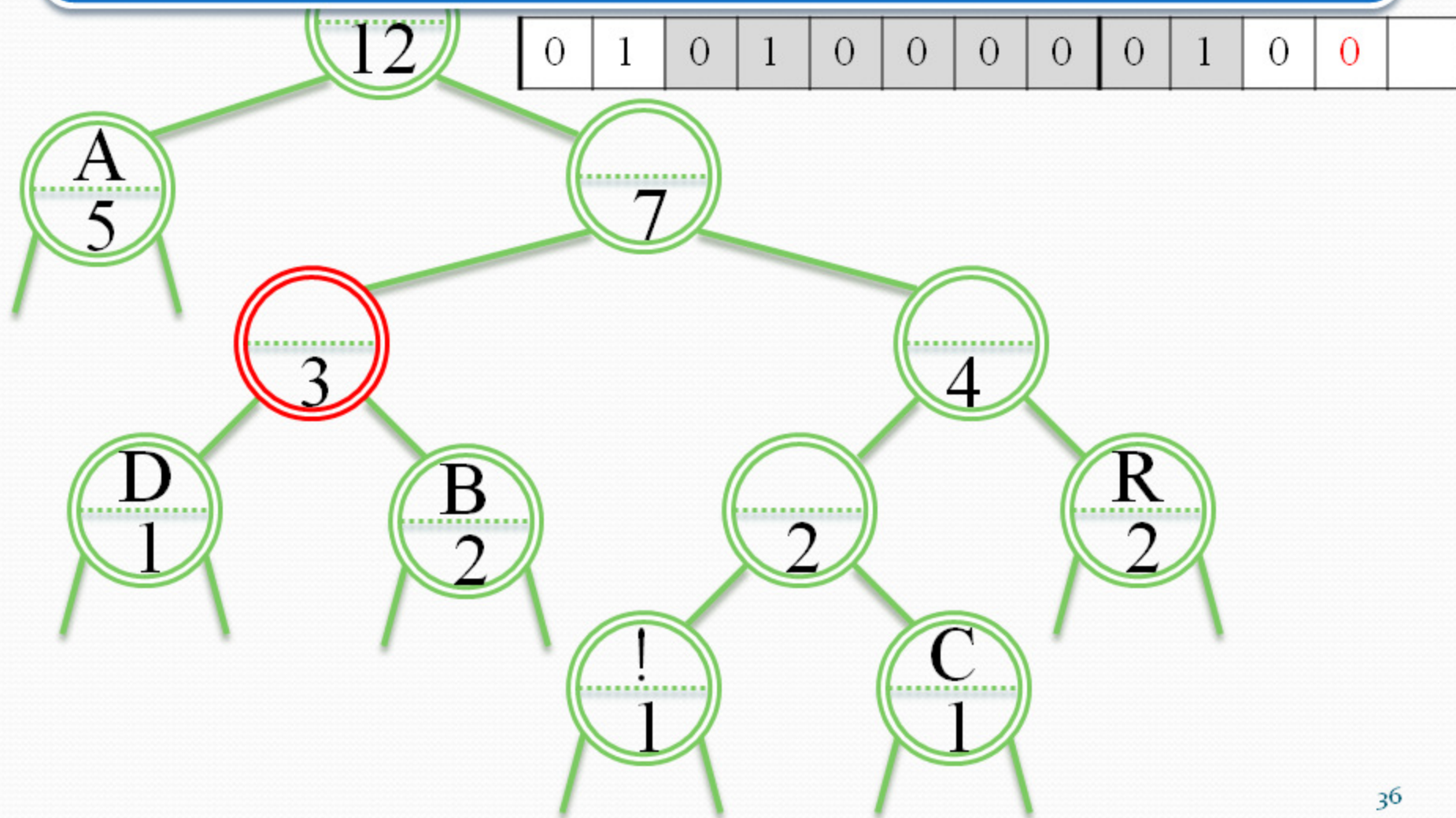
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



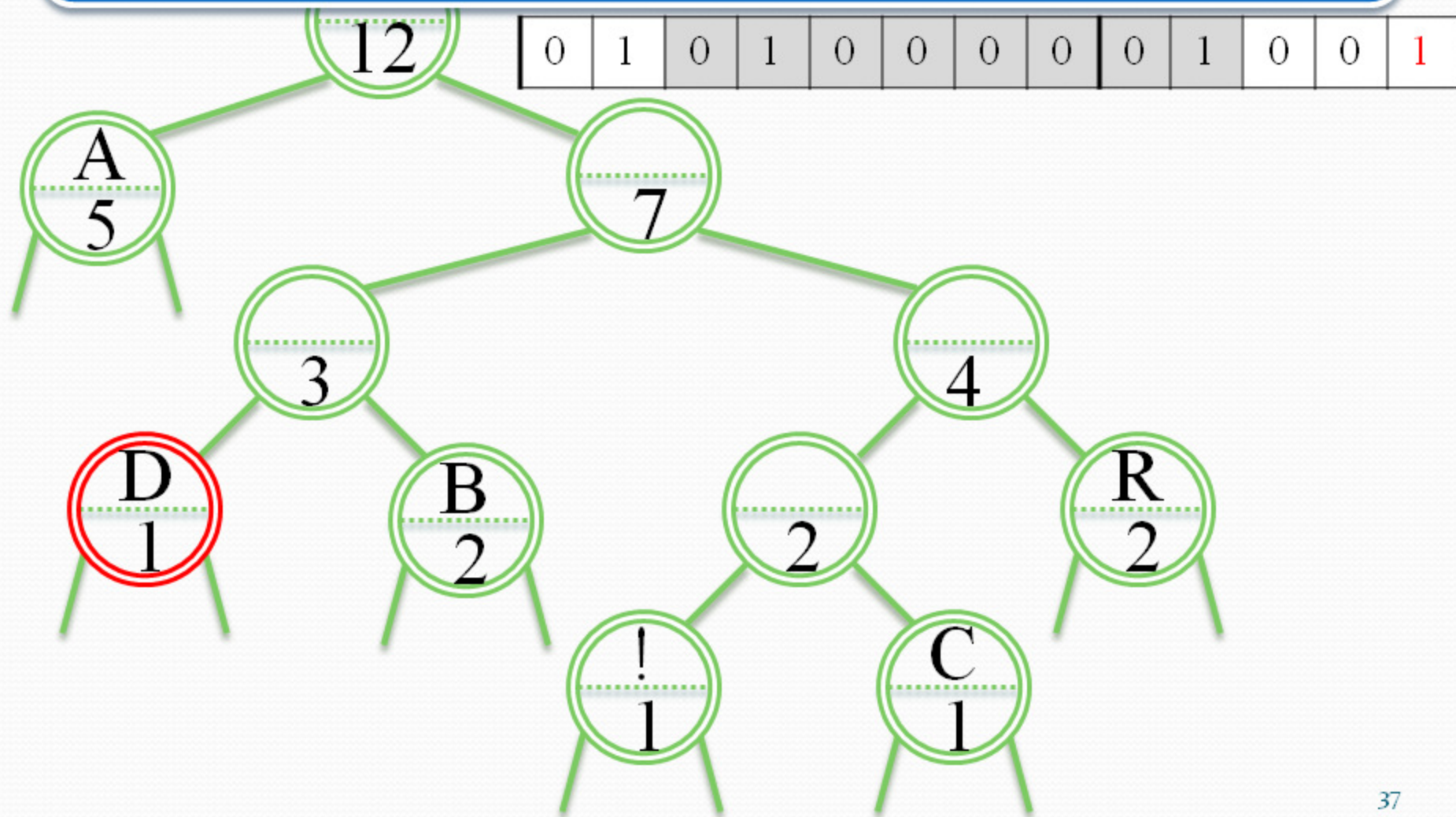
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



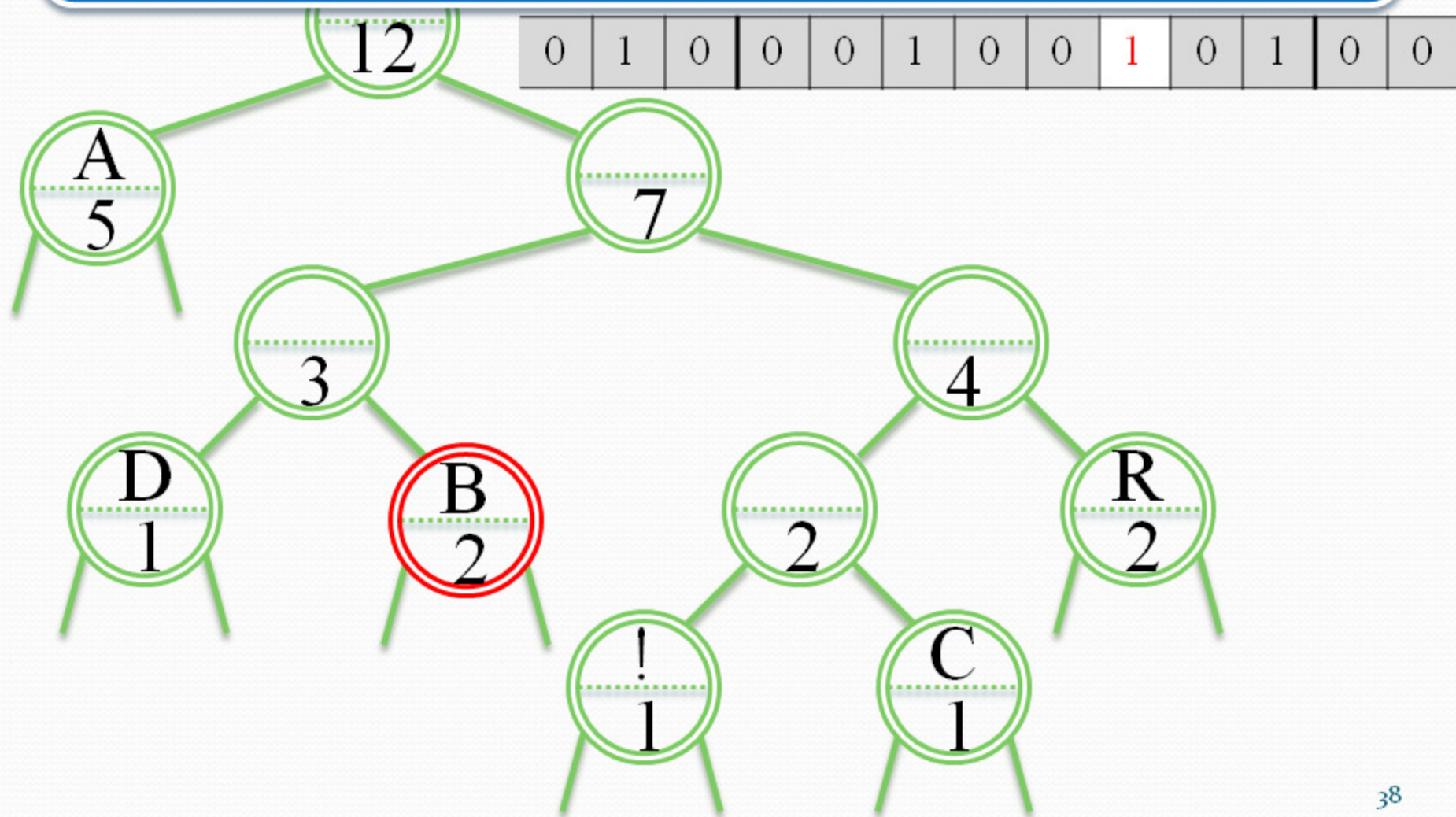
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



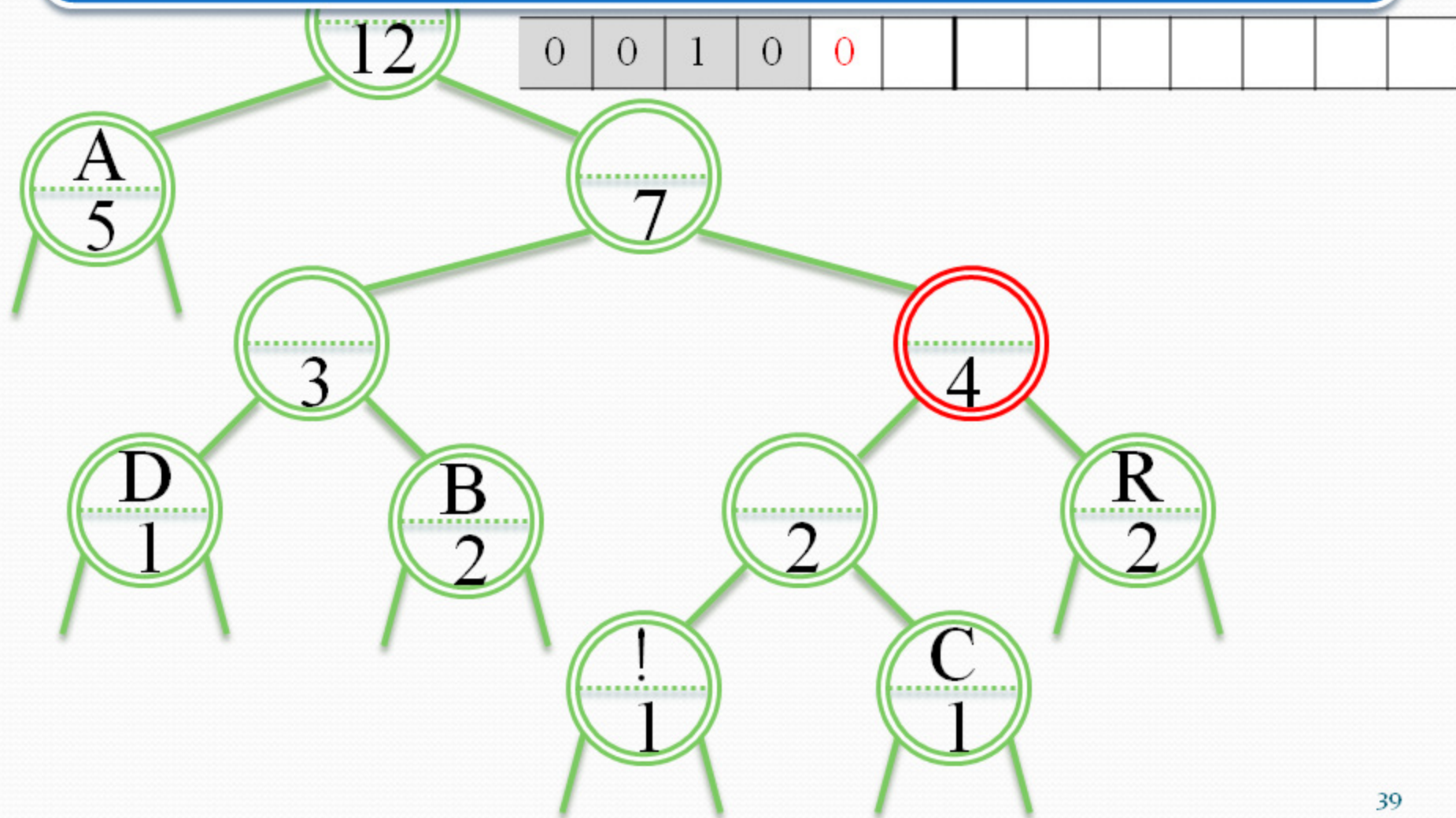
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



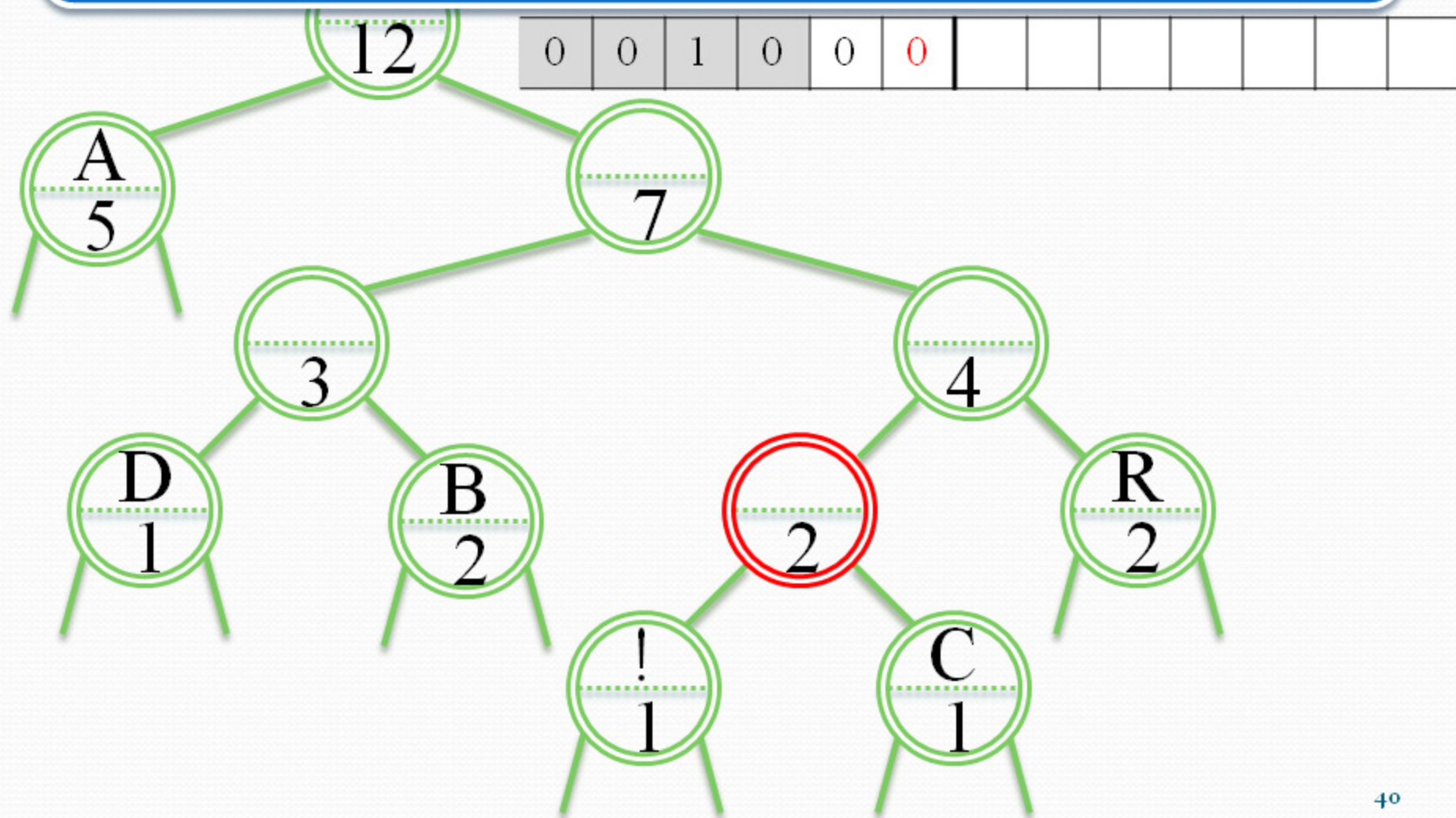
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



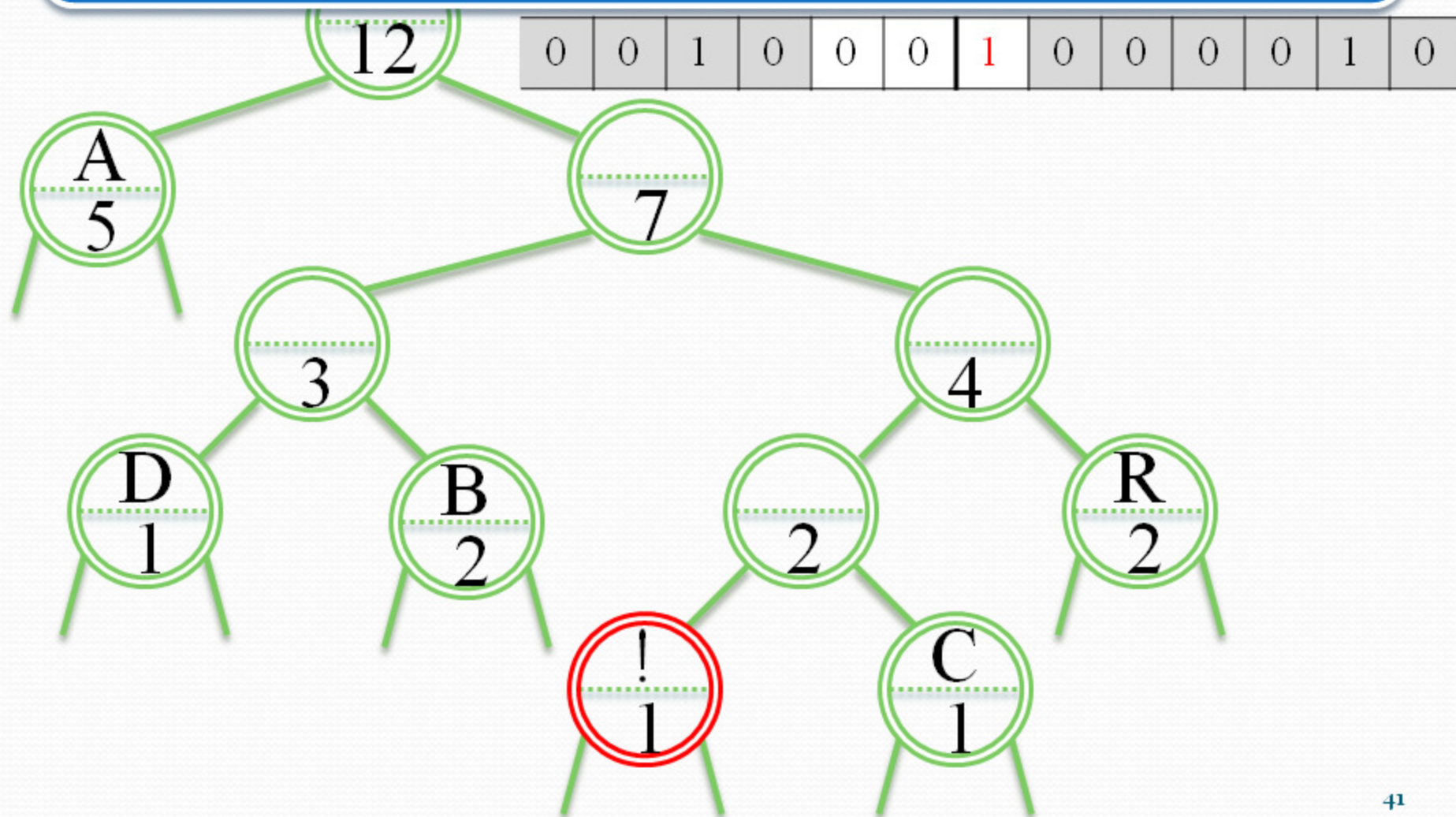
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



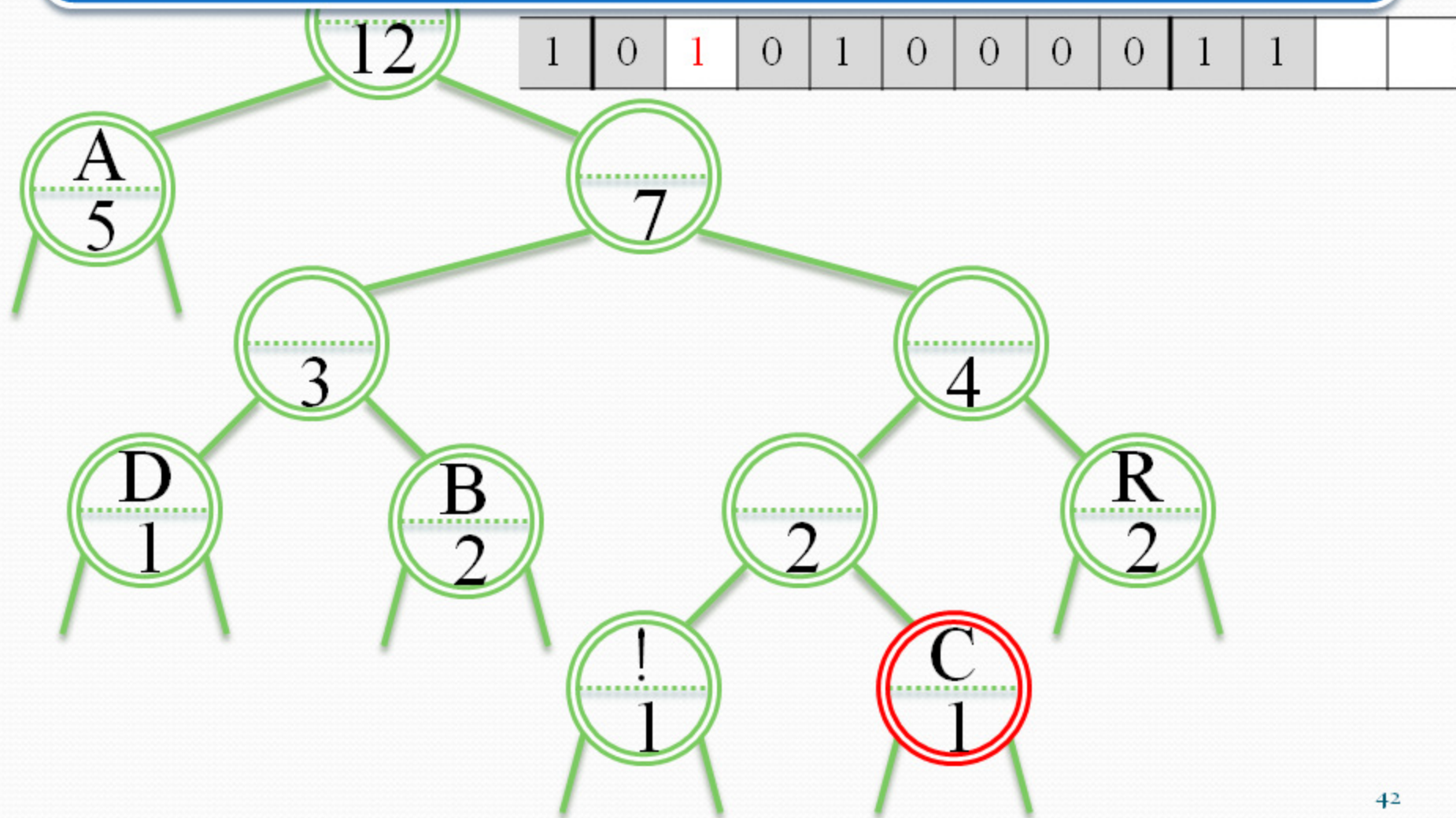
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



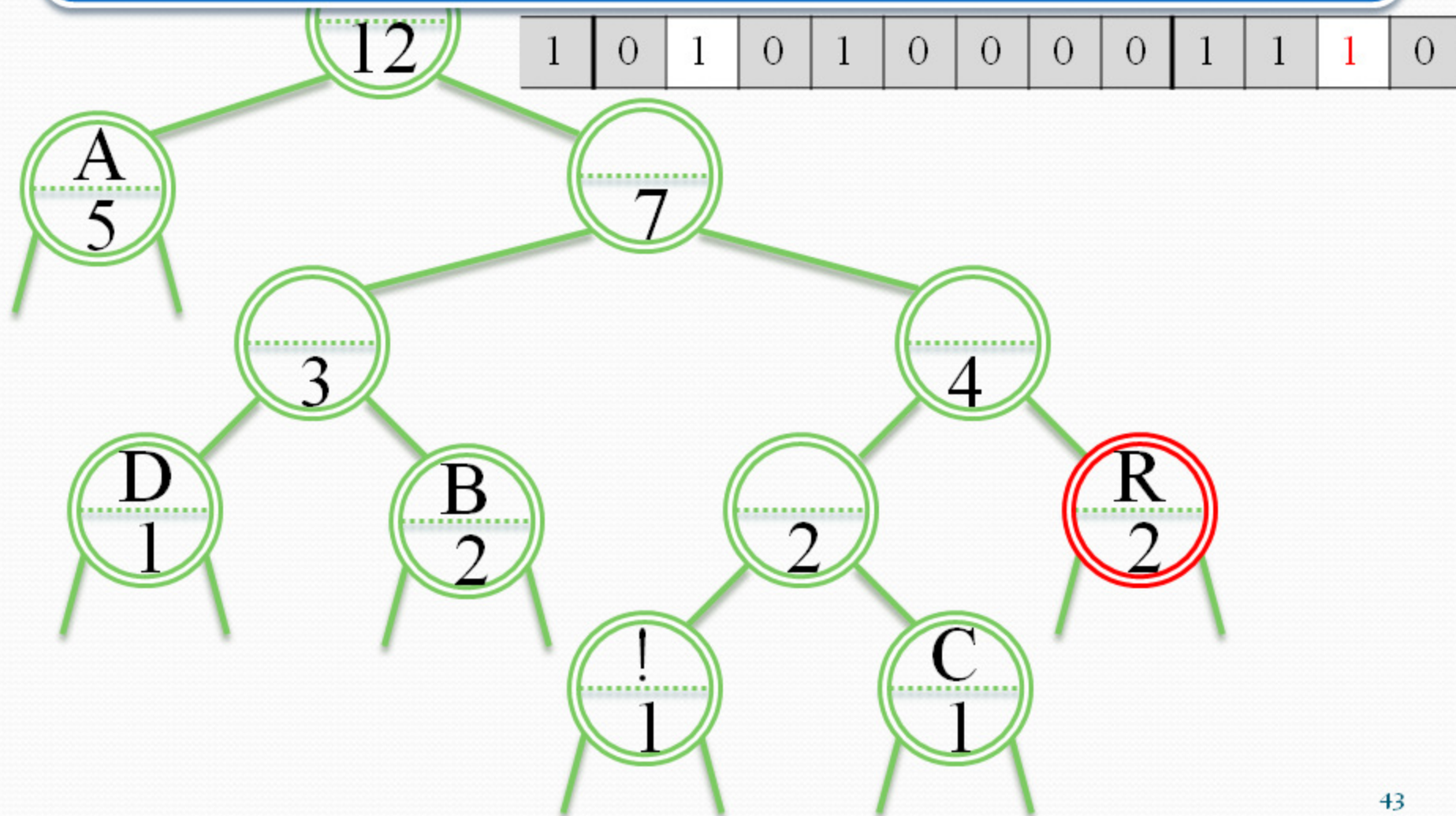
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



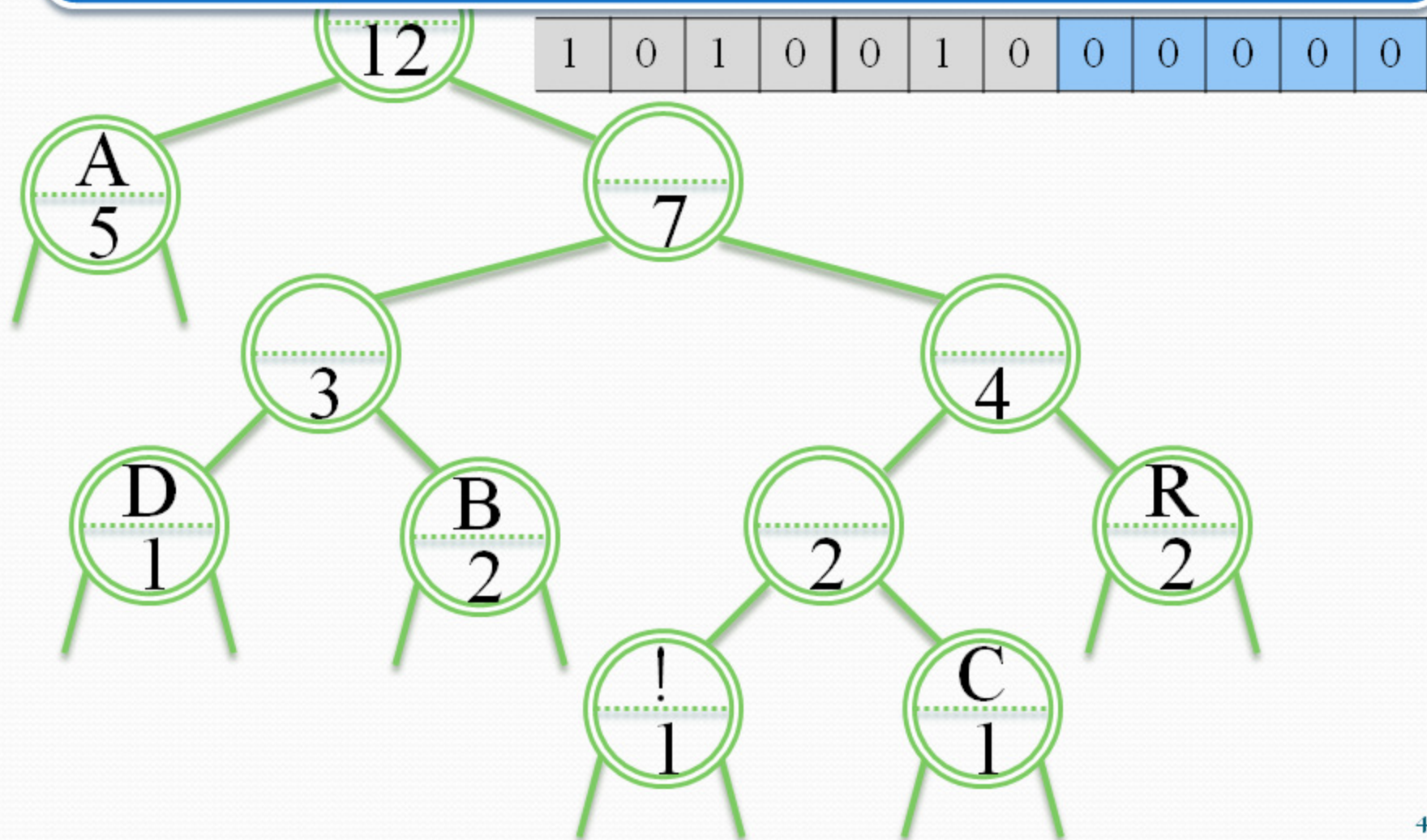
Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



Algorytm Huffmana: kompresja

(5/7) zapis drzewa Huffmana (preorder)



Algorytm Huffmana: kompresja

(6/7) zapis liczby znaków występujących w tekście

1 bajt = 2^8 znaków = 256 B

2 bajty = 2^{16} znaków = 64 KB

3 bajty = 2^{24} znaków = 16 MB

4 bajty = 2^{32} znaków = 4 GB

5 bajtów = 2^{40} znaków = 1 TB

Algorytm Huffmana: kompresja

(7/7) zapis (kompresja) danych wejściowych

ZNAK	SŁOWA KODOWE		Częstotliwość znaku [f]	Głębokość znaku w drzewie
	KODOWANIE ANSI	KOD BEZPREFIKSOWY		
!	00100001	1100	1	4
A	01000001	0	5	1
B	01000010	101	2	3
C	01000011	1101	1	4
D	01000100	100	1	3
R	01010010	111	2	3

$$\text{Koszt kompresji} = \sum_{i=1}^n f_i * \text{głębokość } i - \text{tego znaku}$$

Algorytm Huffmana: kompresja

(7/7) zapis (kompresja) danych wejściowych

Plik	Koszt kompresji * [bajt]	Koszt zapisu drzewa * Huffmana [bajt]
„ABRACADABRA!”	4 (28+4 bitów)	8 (59+5 bitów)
„A TALE OF TWO CITIES”	440702	93
„OPOWIEŚĆ O DWÓCH MIASTACH”	254980	112

* Po wyrównaniu do pełnych bajtów.

Koszt zapisu liczby znaków występujących w pliku: 4 bajty.

Algorytm Huffmana: kompresja

(7/7) zapis (kompresja) danych wejściowych

Plik	Rozmiar pliku		Stopień kompresji [%]
	Przed kompresją [bajt]	Po kompresji [bajt]	
„ABRACADABRA!”	12	16	133.33
„A TALE OF TWO CITIES”	773125	440799	57.02
„OPOWIEŚĆ O DWÓCH MIASTACH”	421008	255096	60.59

Algorytm Huffmana: dekompresja

1. Wczytać dane wejściowe.
 - a) Wczytać zakodowane drzewo Huffmana.
 - b) Wczytać liczbę znaków do odkodowania.
2. Wykorzystać drzewo Huffmana do odkodowania pozostałej części strumienia bitów.

Algorytm Huffmana: dekompresja

(1a/2) wczytanie drzewa Huffmana (preorder)

```
Node* readTree()
{
    if(bit w stumieniu == '1')
    {
        // Znak do przechowania w liściu:
        ch_temp = 'Osiem kolejnych bitów';

        return new Node(ch_temp, 0, NULL, NULL);
    }

    return new Node(0, 0, readTree(), readTree());
}
```

Algorytm Huffmana: dekompresja

(1b/2) wczytanie liczby znaków do zakodowania

Wczytanie 4 bajtów.

Algorytm Huffmana: dekompresja

(2/2) odkodowanie strumienia bitów

