Adatstruktúrák és Algoritmusok

11. gyakorlat

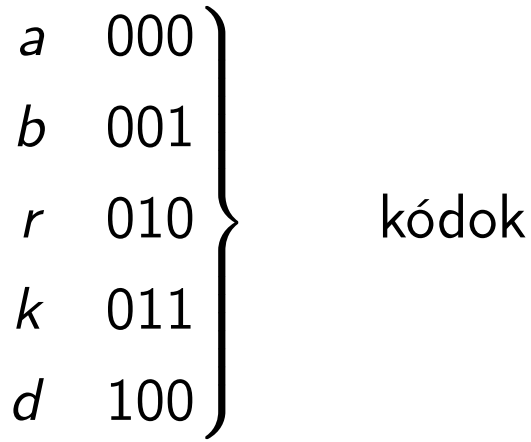
A Huffman-kód

Fix hosszúságú kódok

Szeretnénk kódolni az ABRAKADABRA szót.
Az első lehetőség az, hogy minden karakterhez hozzárendelünk egy fix hosszúságú kódot. Ha k jelöli a kód hosszát, akkor teljesülnie kell a

$$2^{k-1} < 5 \leqslant 2^k \qquad \Longrightarrow \qquad k = \lceil \log(5) \rceil = 3$$

egyenlőtlenségnek. Így kapjuk, hogy 3 hosszúságú kódot kell használnunk.


a 000
b 001
r 010
k 011
d 100
kódok

Kódolás: 000 001 010 000 011 000 100 000 001 010 000
Ez egy 33 bites kódszó. Természetesen nincsenek szóközök, ez csak a könnyebb áttekinthetőség miatt írtam.
A dekódolás rendkívül egyszerű. A kódszót 3 bites részekre kell bontanunk, és könnyen visszakapjuk a kódolt szöveget.

$$\underbrace{000}_{a}\,|\,\underbrace{001}_{b}\,|\,\underbrace{010}_{r}\,|\,\underbrace{000}_{a}\,|\,\underbrace{011}_{k}\,|\,\underbrace{000}_{a}\,|\,\underbrace{100}_{d}\,|\,\underbrace{000}_{a}\,|\,\underbrace{001}_{b}\,|\,\underbrace{010}_{r}\,|\,\underbrace{000}_{a}$$

A Huffman kód

A Huffman-kód elkészítéséhez először is ismernünk kell a karakterek gyakoriságait (illetve relatív gyakoriságait). Ezt követően felépítünk egy bináris fát. A kapott fa fogja szolgáltatni a kódot, illetve ennek a fának a segyítségével tudunk dekódolni.

A faépítés kódolás és dekódolás lépései

1. **Elindulás:** A gyökér legyen a két legkisebb gyakoriságú karakter gyakoriságának az összege. A levelek legyenek a két legkisebb gyakoriságú karakter.

2. **Továbbhaladás:**
   - Már vannak részfáink illetve karaktereink gyakoriságokkal. Ezeket nagyság szerint rendezzük a részfák gyökerében szereplő szám illetve a karakterek gyakoriságai alapján.
   - A kapott sorozat két legkisebb eleméből bináris fát építünk, a kapott gráf gyökerébe a két szám (relatív gyakoriság, vagy gyökérben álló szám) összege kerül.

3. **Leállás:** A rendezések és a faépítések sorozatát addig folytatjuk, míg az összes karakter illetve részgráf elfogy és egyetlen bináris fát alkotnak.

4. **Kiértékelés:** A kapott fagráf éleihez 0-kat és 1-ket rendelünk. A bal oldali gyermekre mutató élhez 0-t, a jobb oldalihoz 1-t rendelünk. A kapott fagráf levelei a karakterek. Minden karakterhez hozzárendelünk egy kódot. Ezt a kódot a gyökértől az adott levélig vezető élsorozat bitjei szolgáltatják.

- **Dekódolás:** A bináris fa ismeretében egyértelmű.

A bináris fa létrehozásánál mindig építünk, aztán rendezünk, aztán megint építünk, megint rendezünk, és így tovább. Gyakori hiba, hogy az építés és rendezésnek ezt a váltakozását nem tartják be. Mi sem írjuk ki mindig, ha a rendezettség nem borul fel.

Érdemes még megjegyezni, hogy a Huffman-kód nem egyértelmű. Például, ha az azonos gyakoriságú kódokat permutáljuk, akkor különböző Huffman kódokat kapunk, azonban ezek a Huffman kódok ugyanarra a szóra azonos hosszúságú kódszavakat eredményeznek.

Feladat

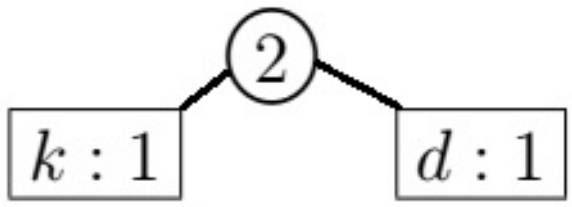
Bár nem életszerű, de a módszert az ABRAKADABRA szó kódolásával illusztráljuk. A gyakoriságokat nyilván nem egy szóból, hanem valamilyen hosszú szövegből kell származtatni, vagy eleve adott.

Megoldás

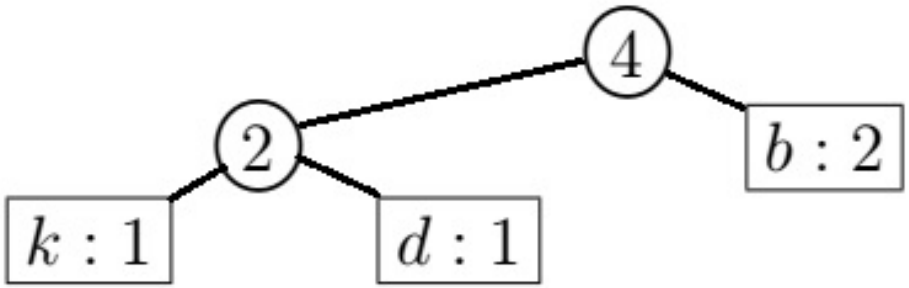
A gyakoriságok:

a	5 db
b	2 db
r	2 db
k	1 db
d	1 db

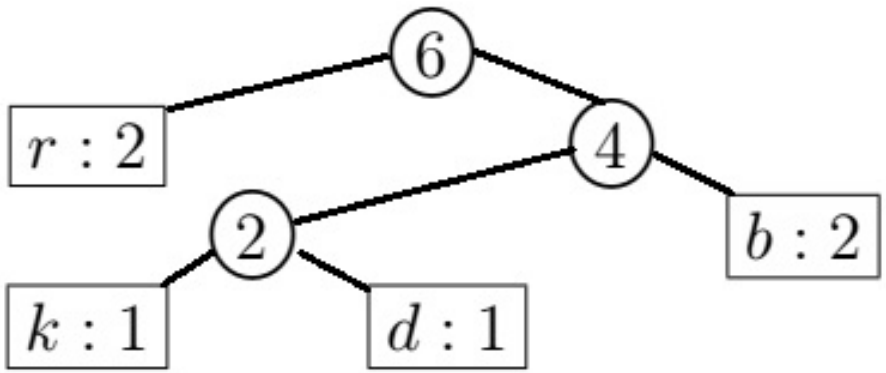
$\underbrace{k:1 \quad d:1}$ $\quad b:2 \quad r:2 \quad a:5$



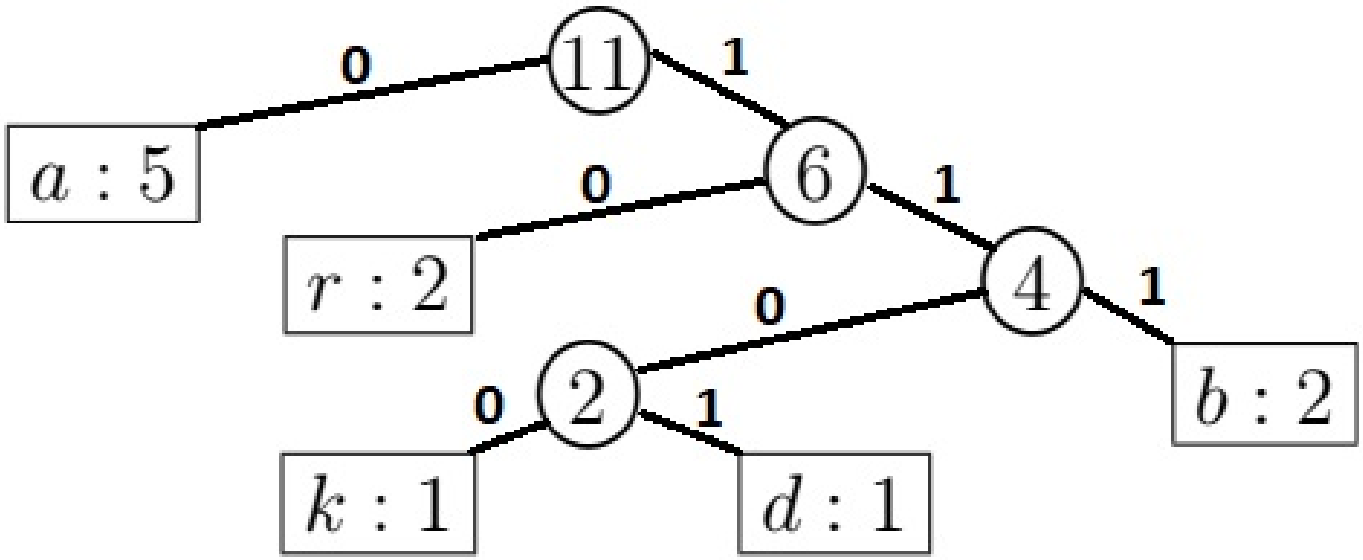
(2) $b:2$ $r:2$ $a:5$



$r : 2$ $\underbrace{\textcircled{4}}$ $a : 5$



$\underbrace{a : 5 \quad \textcircled{6}}$



A kódok

A kapott fagráfot **kódfá**nak nevezzük. A kódfából kiolvashatók a kódok:

a	0
b	111
r	10
k	1100
d	1101

kódok

Kódolás és dekódolás

Kódolás: 0 111 10 0 1100 0 1101 0 111 10 0.
Dekódolás:
A kapott kód prefix kód, tehát mindig egyértelműen dekódolható.

$$\underbrace{0}_{a}|\underbrace{111}_{b}|\underbrace{10}_{r}|\underbrace{0}_{a}|\underbrace{1100}_{k}|\underbrace{0}_{a}|\underbrace{1101}_{d}|\underbrace{0}_{a}|\underbrace{111}_{b}|\underbrace{10}_{r}|\underbrace{0}_{a}$$

Látható, hogy rövidebb (23 bit) kódot kaptunk, mint a fix hosszúságú kódolásnál (33 bit).
A fa költsége optimális.