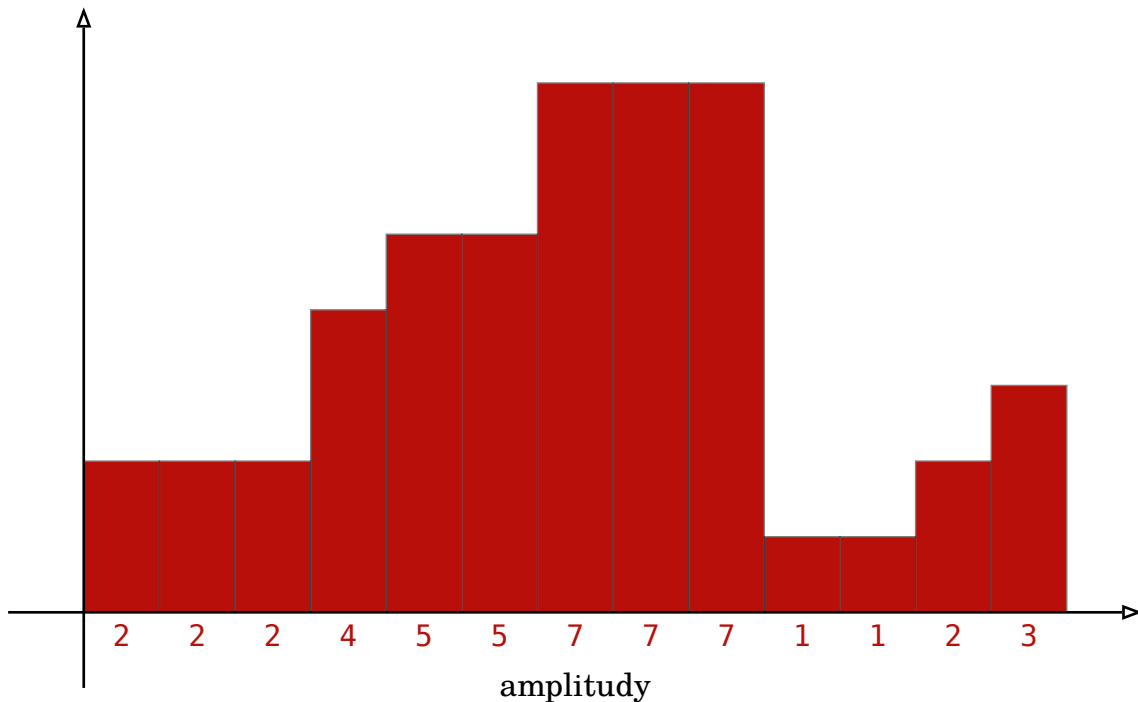


Vzorová zadání lehkých úloh na kompresi

1. Zkomprimujte následující audio (dané jako posloupnost amplitud) použitím **run-length** komprese. Porovnejte velikost komprimovaného audia s původní, když na uložení jedné amplitudy potřebují 2B a délka audia nepřesáhne 1000 amplitud.



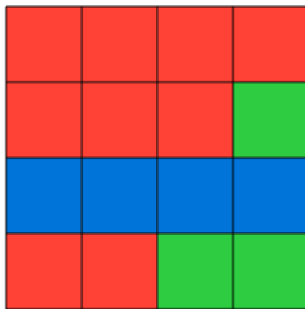
Řešení

Run-length komprese nahradí každou posloupnost amplitud dvojicí čísel (délka posloupnosti, amplituda). V tomto případě nahradíme první trojici dvojek za dvojici (3, 2), následující čtyřku za (1, 4), další dvě pětky za (2, 5) atd. Výsledná posloupnost bude

3 2 1 4 2 5 3 7 2 1 1 2 1 3

Protože na uložení amplitudy potřebují 2B a na uložení délky posloupnosti rovněž 2B ($2^{16} = 65536 > 1000$), bude mít komprimované audio velikost $14 * 2 = 28B$. Původní audio sestávalo z 13 amplitud, takže mělo $13 * 2 = 26B$. V tomto případě byla run-length komprese tak akorát na škodu. RIP.

2. Zkomprimujte následující obrázek použitím (Huffmanova) stromu priorit. Stačí výsledná posloupnost reprezentující obrázek, samotný strom kódovat nemusíte.



Řešení

Nejprve si obrázek překreslíme jako posloupnost.

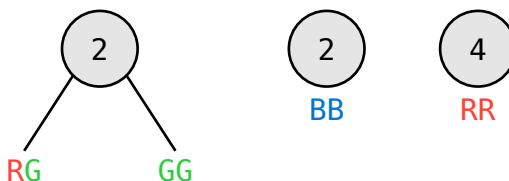


Teď ji rozdělíme na dvojice; různé typy dvojic budou: RR, RG, BB a GG. Dvojice RR je v obrázku 4x, BB 2x a zbylé dvě dvojice jen jednou.

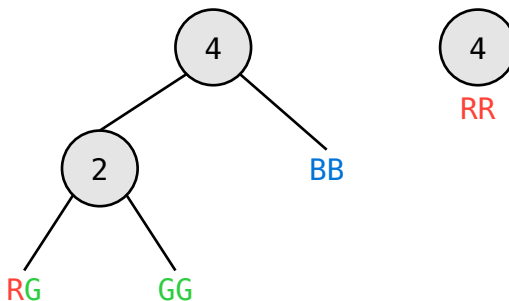
Začneme stavět Huffmanův strom. Ze začátku máme čtyři uzly:

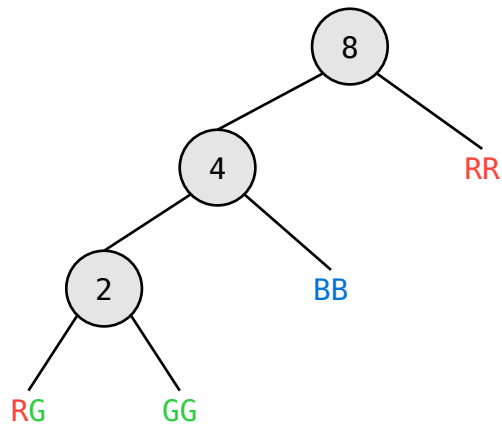


Uzly s nejnižšími četnostmi spojíme ve strom s kořenem držícím součet četností těchto uzlů.



Takhle postupujeme dál, dokud nemáme jediný strom.





Když značíme „pohyb“ doleva jako 0 a doprava jako 1, dostaneme následující kódování.

| blok | kód |
|-------------|------------|
| RR | 1 |
| BB | 01 |
| GG | 001 |
| RG | 000 |

Konečná posloupnost je tedy

1 1 1 000 01 01 1 001.