

Huffman codes for Data compression

You are given a file text characters appearing in some frequency.

Let $B(T)$ = size of encoded file

$$= \sum \text{char } f(c) d(c)$$

= # bits needed

Problem: You want to minimize $B(T)$, i.e.: # bits to encode file

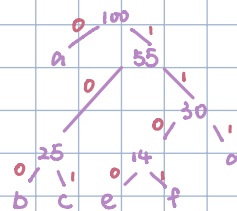
char	type	a	b	c	d	e	f
$f(c)$	Frequency	45	13	12	16	9	5
$d(c_1)$	fixed size	000	001	010	011	100	101
$d(c_2)$	var size	0	101	100	11	1101	1100

→ is it good encoding? X \Rightarrow 011 000 011 \Rightarrow 9 bits
 \overline{dad}
 1110111
 ↑ ↑ ↑
 a a d
 → read through, if found then translate \Rightarrow 7 bits

Greedy algorithm:

Repeat. Unite two smallest frequency keys building a binary tree by summing their frequencies

char	type	a	b	c	d	e	f
$f(c)$	Frequency	45	13	12	16	9	5
			25	30	14		
				55			
i.e.:		dad 1110111	100				
$d(c)$	var size	0	100	101	11	1100	1101



To implement: Use a min-heap to extract the min

Time: $O(n \log n)$