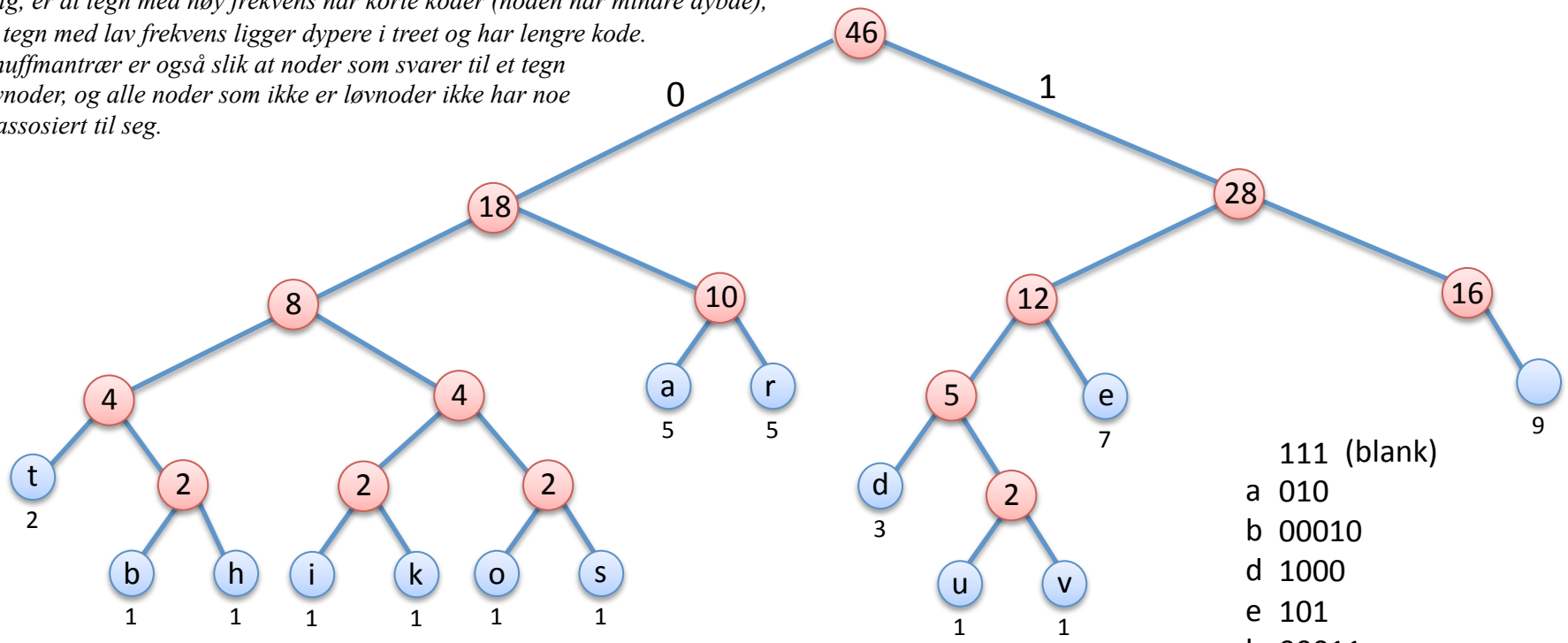
Når treet bygges brukes en prioritetskø, jf. algoritmen i G&T.
removeMin() kan fjerne noder som har samme frekvens i vilkårlig rekkefølge.
Det kan derfor finnes mange 'riktige' huffmantrær for en gitt oppgave.
Dette gjelder begge oppgavene her. En test på om løsningsforslaget er
rimelig, er at tegn med høy frekvens har korte koder (noden har mindre dybde),
og at tegn med lav frekvens ligger dypere i treet og har lengre kode.
Alle huffmantrær er også slik at noder som svarer til et tegn
er løvnoder, og alle noder som ikke er løvnoder ikke har noe
tegn assosiert til seg.

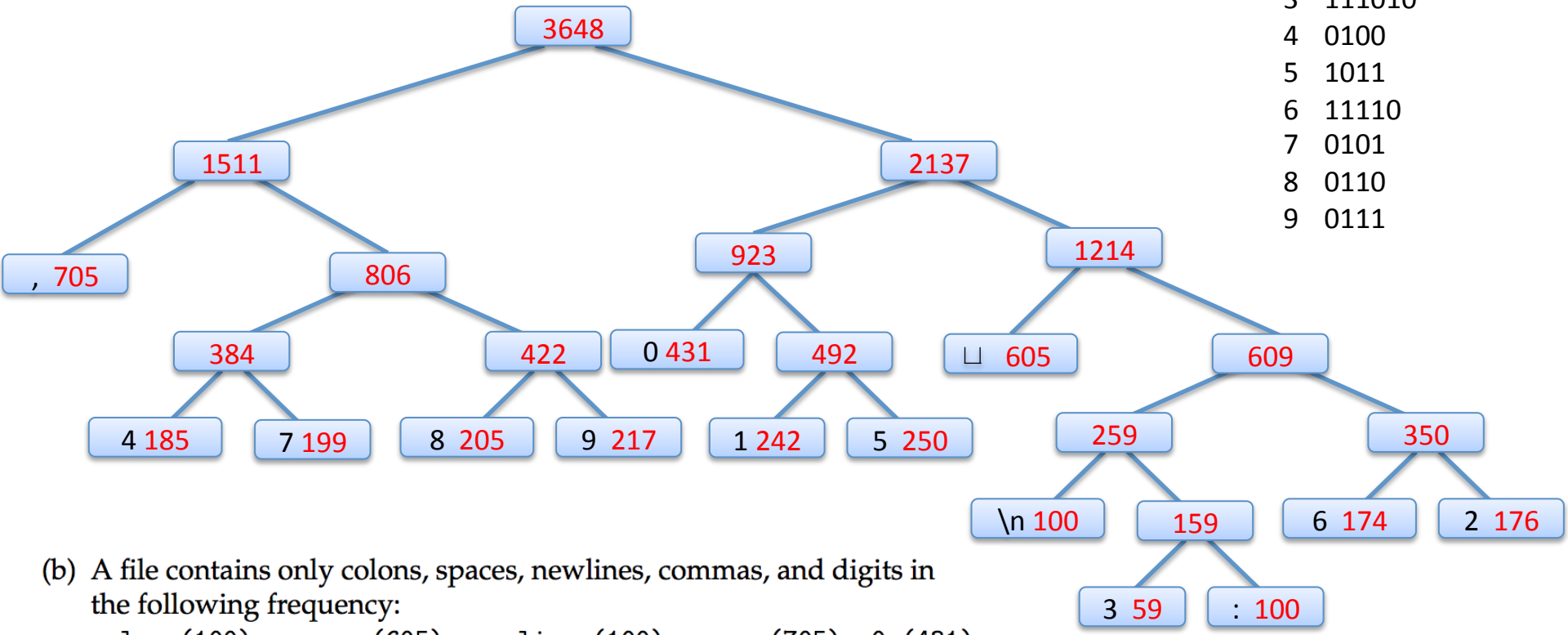


	111 (blank)
a	010
b	00010
d	1000
e	101
h	00011
i	00100
k	00101
o	00110
r	011
s	00111
t	0000
u	10010
v	10011

(a) A file contains only spaces and digits in the following frequency:
`space (9), a (5), b (1), d (3), e (7), f (3), h (1), i (1), k (1), n (4), o (1), r (5), s (1), t (2), u (1), v (1).`
Construct the Huffman code.

Når treet bygges brukes en prioritetskø, jf. algoritmen i G&T.
removeMin() kan fjerne noder som har samme frekvens i vilkårlig rekkefølge.
Det kan derfor finnes mange 'riktige' huffmantrær for en gitt oppgave.
Dette gjelder begge oppgavene her. En test på om løsningsforslaget er
rimelig, er at tegn med høy frekvens har korte koder (noden har mindre dybde),
og at tegn med lav frekvens ligger dypere i treet og har lengre kode.
Alle huffmantrær er også slik at noder som svarer til et tegn
er løvnoder, og alle noder som ikke er løvnoder ikke har noe
tegn assosiert til seg.

,	00
␣	110 (blank)
\n	11100
:	111011
0	100
1	1010
2	11111
3	111010
4	0100
5	1011
6	11110
7	0101
8	0110
9	0111



(b) A file contains only colons, spaces, newlines, commas, and digits in the following frequency:
`colon (100), space (605), newline (100), comma(705), 0 (431), 1 (242), 2 (176), 3 (59), 4 (185), 5 (250), 6 (174), 7 (199), 8 (205), 9 (217).`
Construct the Huffman code.