

Checkpoint 1:

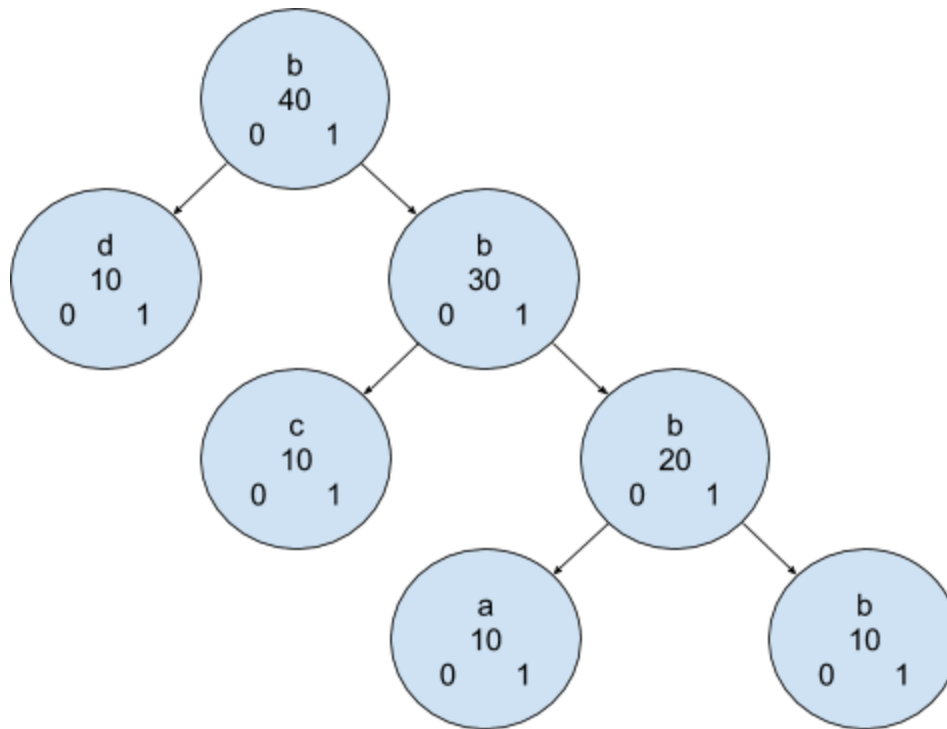
First we build the tree with:

a: 10

b: 10

c: 10

d: 10



Decoded as:

a:110 b: 111 c: 10 d:0

abcd = 110111100,

Expected output:

$$abcd \cdot 10 =$$
[illegible]

Output from my compress:

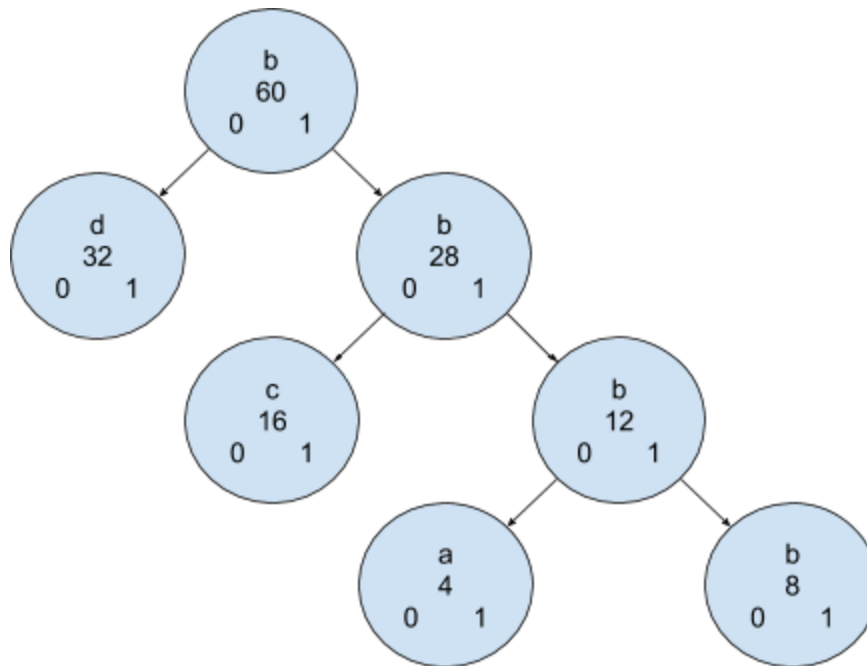
```
1101111001101111001101111001101111001101111001101111001101111001101111001101  
11100110111100
```

Which is identical with the output of my compressed file. Nothing to be fixed.

Checkpoint 2:

First we build the tree with:

- a: 4
- b: 8
- c: 16
- d: 32



Decoded as:

a:110 b: 111 c: 10 d:0

aa = 110110

bbbb = 111111111111

ccccccc = 1010101010101010

dddddddddddddd = 0000000000000000

aabbbbccccccddddd=11011011111111111110101010101010100000000000000000000

00
ddddddddddccccccbbbaa=0000000000000000101010101010101011111111111110110

Expected output:

11011011111111111110101010101010100000000000000000000000000000001010101010101011111111111110110

Output from my compress:

11011011111111111110101010101010101000000000000000000000000000001010101010101011111111111110110

Which is identical with the output of my compressed file. Nothing to be fixed.

```
[p2chu@ieng6-201]:PA3:537$ cat input1.txt
abcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
[p2chu@ieng6-201]:PA3:538$ cat checkpoint1.txt
abcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
```

```
[p2chu@ieng6-201]:PA3:535$ cat input2.txt
aabbbbcddddddddddddddddddddddddddddddccccccbbbbaa
[p2chu@ieng6-201]:PA3:536$ cat checkpoint2.txt
aabbbbcddddddddddddddddddddddddddddddccccccbbbbaa
```

Checkpoint1.output:

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Checkpoint2.output.txt:

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]