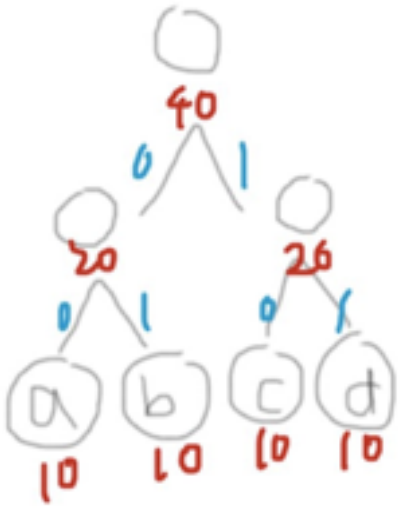
Timothy Koh
Madeline Hsia
File: checkpoint.pdf

checkpoint1.txt
abcdabcdabcdabcdabcdabcdabcdabcdabcdabcd

out.txt:
00011011000110110001101100011011000110110001101100011011000110110001101100011011

1. Start with 4 nodes in the forest, a,b,c,d with the same frequency
2. Since they're the same frequency, compare symbols (return symbol > other.symbol).
3. Tree gets constructed from left to right as shown in the photo below
4. To find the code word for each character, follow the tree from top to bottom, starting at the root



Encoding
a: 00
b: 01
c: 10
d: 11

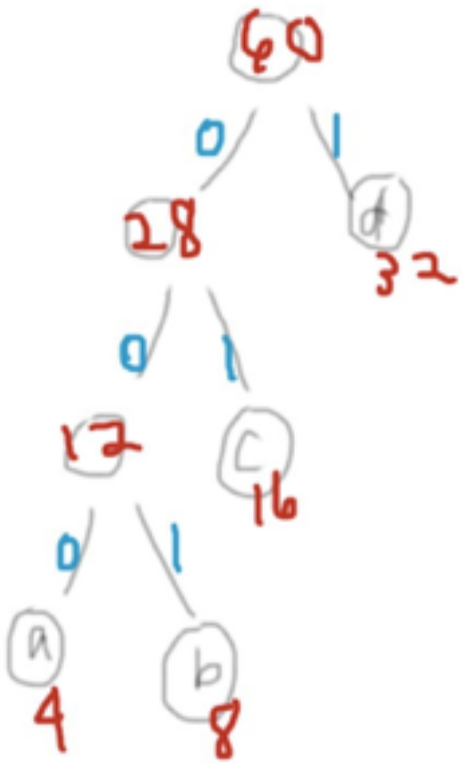
Frequencies
a: 10
b: 10
c: 10
d: 10

Manually encoded string would be equal to the output.
00011011000110110001101100011011000110110001101100011011000110110001101100011011

The problem above was encountered at first when the handwritten tree showed the 00,01,10,11 encoding, but the output gave 11,10,01,00 (d,c,b,a), but this was easily fixed by changing return symbol < other.symbol. It does not really matter which way you encode as long as it is consistent, but it makes more sense for it to go from left to right.

checkpoint2.txt
aabbbbccccccccddddddddddddddddddddddddddddddddddddddddcccccccccbbbbaa

out2.txt
00000000100100100101010101010101011111111111111111111111111111111101010101010101010
01001001001000000



Encoding
a:000
b:001
c:01
d:1

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

1. Each have different frequency, so first find the two smallest (a and b)
2. Combine and join with new parent node (12)
3. Remove the two, and combine 12 and c(16) to new parent node (28)
4. Remove 12 and c, combining (28) and d(32) to the new node (60)
5. Now the queue is empty, so set 60 as root

Manually encoded string would be equal to the output.
00000000100100100101010101010101011111111111111111111111111111111101010101010101010
01001001001000000