CIS 3223 HW 9 | **Name:** Solutions

Dr Anthony Hughes | **Temple ID (last 4 digits:**

1 (24 pts) Consider the following letter frequency table.

Letter	A	B	D	E	G	L	R	U
Frequency	31	15	32	70	10	15	20	7
	6	3	7	8	2	4	5	1

200

Construct the Huffman tree.

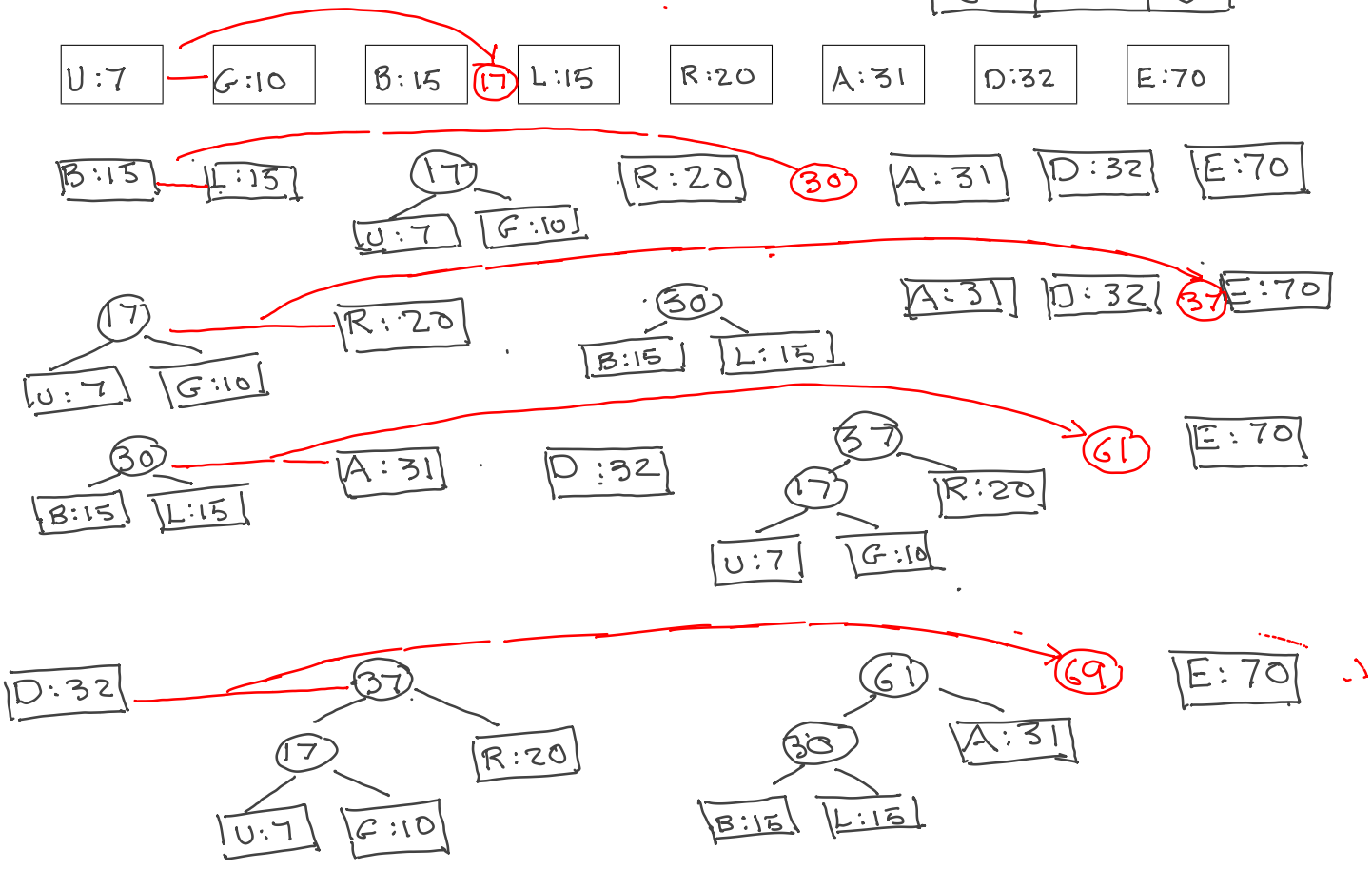
Complete the table, list letters in **binary order**.

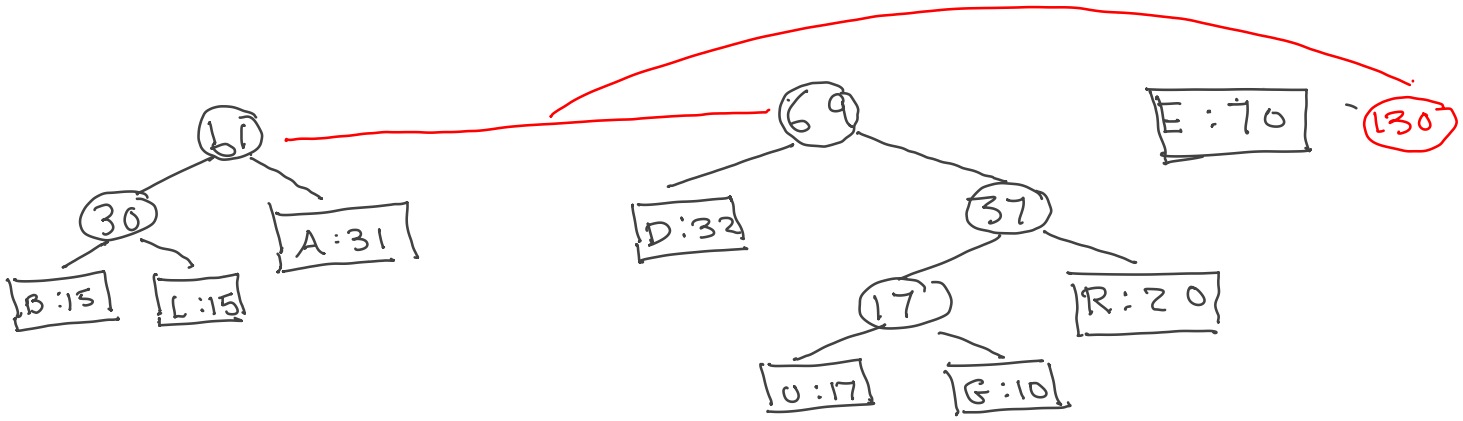
Letter	Code	Bits
E	0	1
A	101	3
D	110	3
B	1000	4
L	1001	4
R	1111	4
U	11100	5
G	11101	5

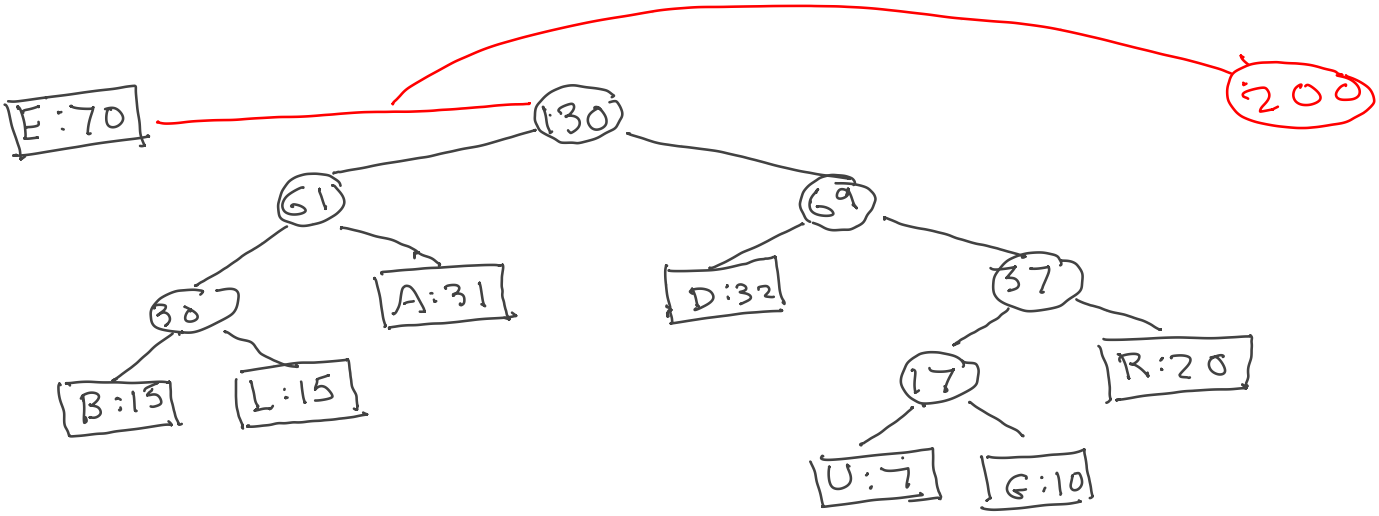
Average number of bits per letter 2.72

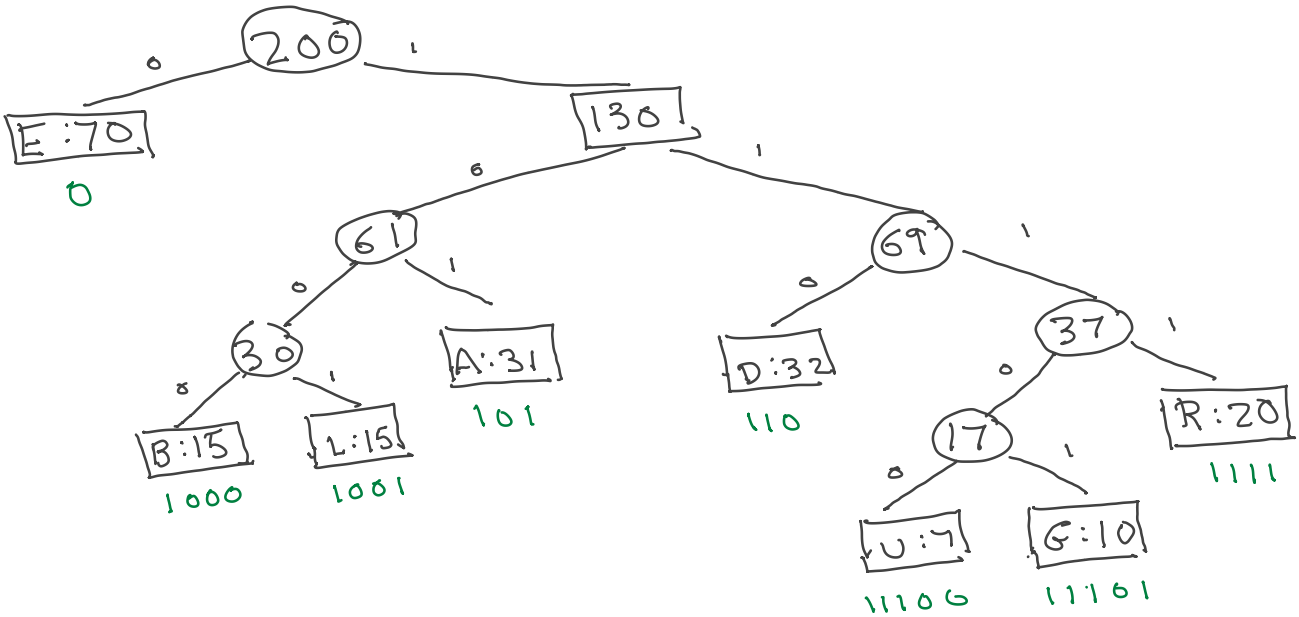
Encode: BREAD 1000111101011110

Decode: 1111|11100|11101|11101|0|110 RUGGED

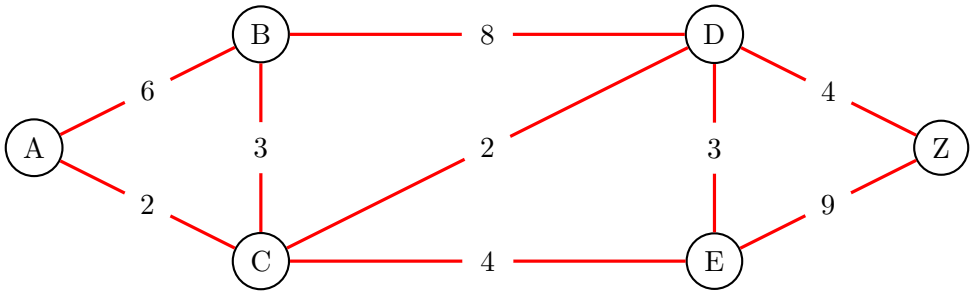



61
69
E:70
130
30
A:31
D:32
37
B:15
L:15
17
R:20
U:7
G:10


200
E:70
130
61
69
30
A:31
D:32
37
B:15
L:15
17
R:20
U:7
G:10


200
0
1
E:70
130
0
1
61
69
0
1
0
1
30
A:31
D:32
37
0
1
0
1
B:15
L:15
17
R:20
0
1
0
1
U:7
G:10
1000
1001
101
110
1110
1111
1111

2 (16 pts) Consider the following undirected graph:



Running Dijkstra's algorithm we have the following data after `deletemin` → C

parent	A	A	A			
S	1	0	1	0	0	0
dist	0	6	2	∞	∞	∞
pos	-1	1	-1	4	3	2
vertex	A	B	C	D	E	Z

H	B	Z	E	D	4

neighbors of C = {A, B, D, E }.

Execute the next two `updates`.

B: dist(C) + w(C,B) = 2+3 = 5 < 6
update dist(B)

B 6 5
Z ∞
E ∞
D ∞

no change

parent	A	C	A			
S	1	0	1	0	0	0
dist	0	5	2	∞	∞	∞
pos	-1	1	-1	4	3	2
vertex	A	B	C	D	E	Z

H	B	Z	E	D	4

D: dist(C) + w(C,D) = 2+2 = 4 < ∞
update dist(D)

B 5
Z ∞
E ∞
D ∞ 4

D 4
B 5
E ∞
Z ∞

pos
H

parent	A	C	A	C		
S	1	0	1	0	0	0
dist	0	5	2	4	∞	∞
pos	-1	2	-1	1	3	4
vertex	A	B	C	D	E	Z

H	D	B	E	Z	4

3 (10 pts) For the Huffman encoding algorithm, use pseudo-code to implement the following.

(a) Nodes of the Huffman prefix tree.

The initial nodes store a symbol and its frequency, these are the leaf nodes. The two nodes with the lowest frequencies are merged into a new node which stores their sum and references to the two merged nodes, these nodes are the intermediate nodes. Indicate how the values stored in each node can be retrieved.

(b) Retrieval and insertion.

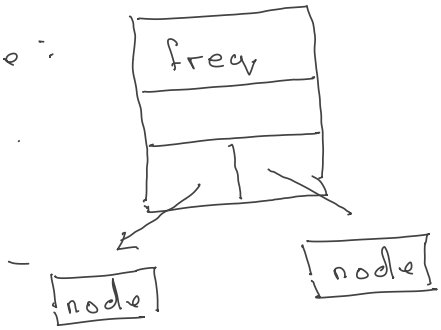
Give the data structure for the storage of the nodes and how the two nodes with the lowest frequencies are retrieved, merged and their merged node inserted.

(a) node A A has fields freq : int
letter : char
left : node
right : node

Access using dot notation

leaf node :

internal node :



(b) Priority queue or Fibonacci queue with freq as key.