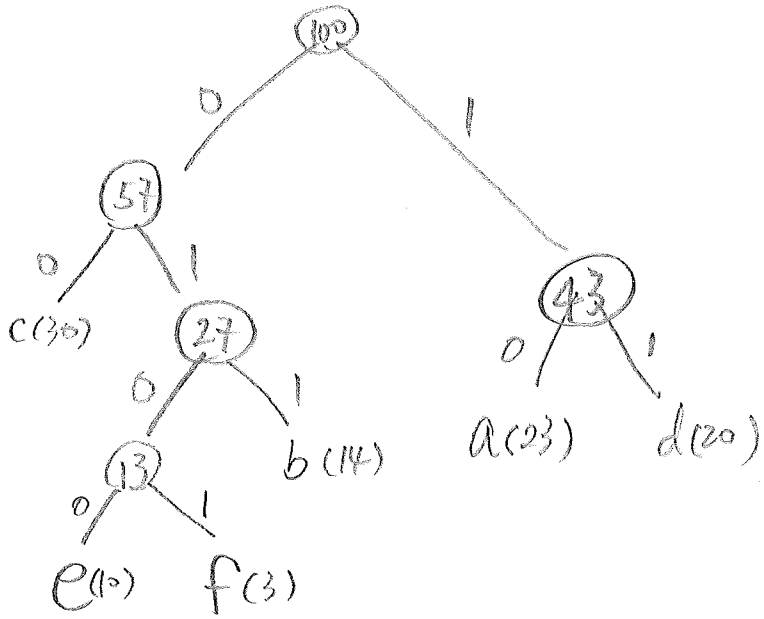
CS325 (Winter 2017) Quiz 4

ID: _____ Name:_____

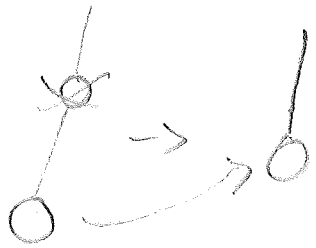
1. (6 pts) Apply the Huffman algorithm to the following alphabet. Please show you binary tree and clearly mark each edge with 0 or 1 and each leave with its character. Also please fill in the table with the final codes.

char.	frequency	code
a	23	10
b	14	011
c	30	00
d	20	11
e	10	0100
f	3	0101



2. (5 pts) Prove that the binary tree representing an optimal prefix-free code must be full.
Definition: A full binary tree is a tree whose non-leave nodes always have two children.

Assume we have a tree T not full, then at least one non-leave node has only one child, like shown in picture. We can delete the non-leave node, this will lead to cheaper code for that leave while all others remain the same. It thus follows that an optimal coding tree must be full, otherwise we will always be able to improve it.



3. *If a graph G has a unique least expensive edge e, then e must be part of every minimum spanning tree of G.*
Consider the following proof to the above statement:

> Assume (for the sake of contradiction) there exists an MST T that does not contain e. We can take an edge f in T that is more expensive than e and replace f with e, which will give us a cheaper spanning tree, leading to a contradiction. Thus every MST must contain e.

(4 pts) What is wrong with this proof? (No need to fix the proof, just point out the bug)

The key issue is that replacing f with e may not lead to a spanning tree.

Extra:

To fix this, we will need to identify specific edge f so that we can guarantee a spanning tree when f is replaced with e.

To do so, we can add e to T. This will create a cycle involving e. Simply deleting an edge other than e from this cycle will create a tree cheaper than T.