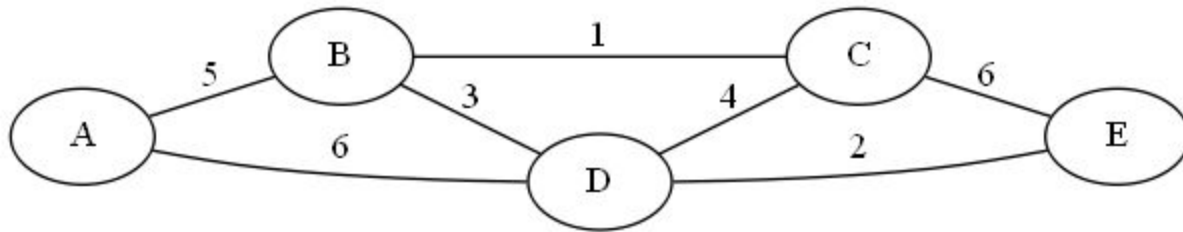Name: Susmitha Shailesh   Date: 11/21/19

Pledge: I pledge my honor that I have abided by the Stevens Honor System.

1.

Apply Kruskal's algorithm to find the minimum spanning tree. Use `makeset(x)`, `find(x)`, and `union(x, y)` to determine if there are cycles. Using the forest representation of subsets taught in class, show the tree after **all** edges have been processed. When processing edges, make sure lower letters appear first. (i.e. use AB, not BA) When drawing the tree, put lower letters in left subtrees so that all the vertices in a level are sorted alphabetically from left to right.

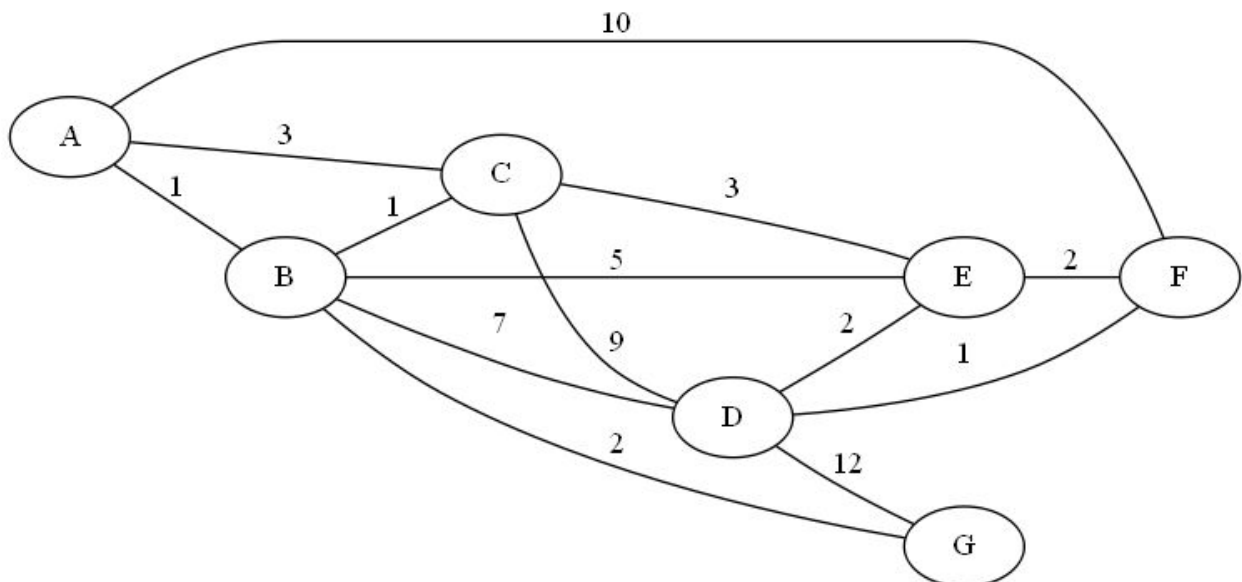| Edges	(BC, 1)	(DE, 2)	(BD, 3)	(CD, 4)	(AB, 5)	(AD, 6)	(CE, 6)
Find(x), Find(y)	B, C	D, E	B, D	B, B	A, B	A, A	A, A
Add?	yes	yes	yes	no	yes	no	no
Edges	C -> B	C -> B E -> D	C -> B D -> B E -> D	C -> B D -> B E -> D	C -> B D -> B E -> D B -> A	C -> B D -> B E -> D B -> A	C -> B D -> B E -> D B -> A

Edges in MST: (A, B), (B, C), (B,D), (D, E)

Weight of MST: 11

Final tree resulting from union-find algorithm:

A
|
B
/ \
C   D
       \
        E
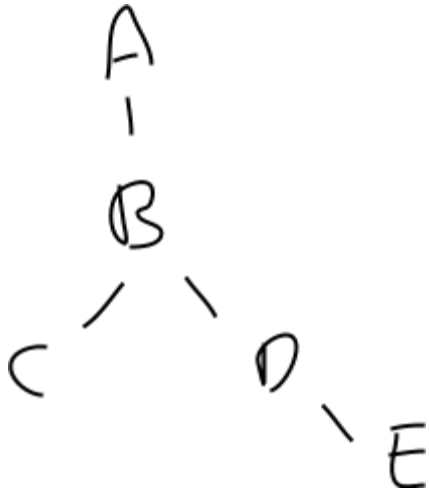
2. Use Dijkstra's algorithm to find the shortest path between nodes **A** and **F**. (Start at A, end at F)

Tree Vertices	Remaining Vertices
A	**B(A, 1)**, C(A, 3), D(-, inf), E(-, inf), F(A, 10), G(-, inf)
B(A, 1)	**C(B, 2)**, D(B, 8), E(B, 6), F(A, 10), G(B, 3)
C(B, 2)	D(B, 8), E(C, 5), F(A, 10), **G(B, 3)**
G(B, 3)	D(B, 8), **E(C, 5)**, F(A, 10)
E(C, 5)	**D(E, 7)**, F(E, 7)
D(E, 7)	**F(E, 7)**
F(E, 7)	Stop

Path: A, B, C, E, F

Distance: 7

3.

a. Construct a Huffman tree for the following data:

Symbol	A	B	C	D	_
Frequency	0.4	0.1	0.2	0.15	0.15
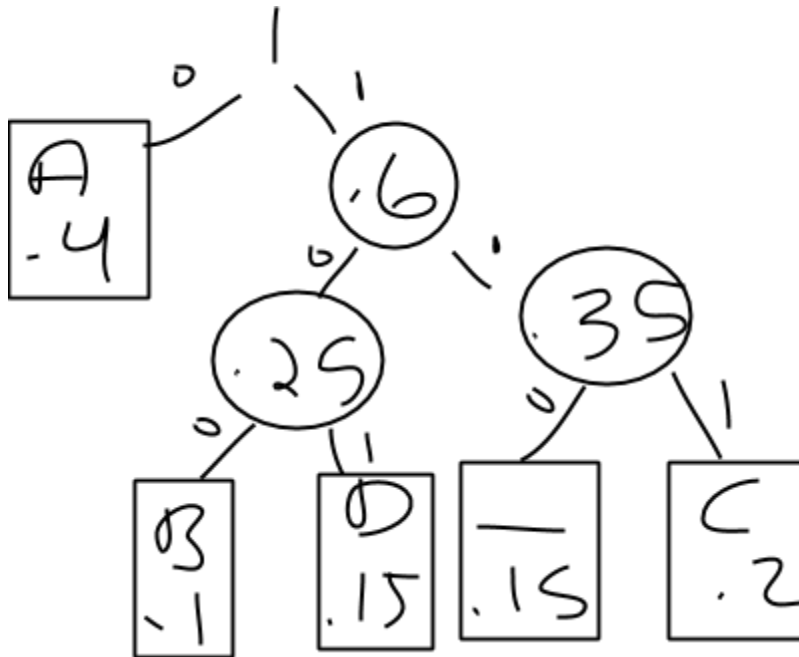
b. Encode ABACABAD using the tree you generated for (a).

0100011101000101

c. Decode 100010111001010 using the tree you generated for (a).

BAD_ADA

d. What compression gain (percent of improvement) do we get by using Huffman encoding instead of a fixed-length encoding scheme (assume the fixed-length scheme would require 3 bits for each character)?

(3 - .4(1) - .1(3) - .15(3) - .15(3) - .2(3)) / 3

26.67%