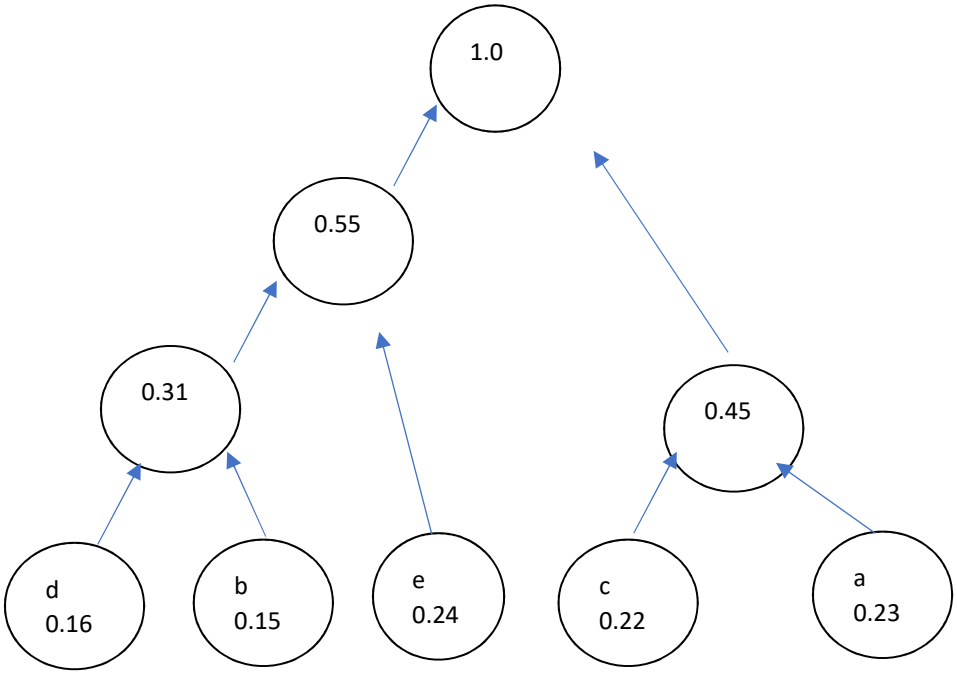
Name and Username: Dinh Nguyen (dln45)

Question 2

For every left edge, put a 0 on the edge. For every right edge, put a 1 on the edge.



So for (a), (b), (c), (d) and (e), the binary code generated are:

Letter	Code
a	11
b	001
c	10
d	000
e	01

Question 3: Start node 2

a) (2,1) with weight = 3
b) (1,0) with weight = 2
c) (1,4) with weight = 6
d) (0,3) with weight = 8
e) Total weight of MST = 3+2+6+8 = 19

Question 4:

(a) (2 points) $T(n) = 4T(n/2) + n^3$
(b) (2 points) $T(n) = 27T(n/3) + n$
(c) (2 points) $T(n) = 8T(n/2) + 1$
(d) (2 points) $T(n) = 25T(n/5) + n^2$
(e) (2 points) $T(n) = 2T(n/2) + n\log_2(n)$

a) c = log2(4) = 2, f(n) = n^3. We find that $n^3 = \Omega(n^{2+e})$ where e = 1. So Case 3 selected, T(n) = $\Theta(n^3)$
b) c = log3(27) = 3, f(n)=n. We find that $n = O(n^{3-e})$ where e = 2. So Case 1 selected, T(n) = $\Theta(n^3)$
c) c=log2(8)=3, f(n) = 1. We find that $1 = O(n^{3-e})$ where e = 3. So Case 1 selected, T(n) = $\Theta(n^3)$
d) c=log5(25)=2, f(n)=n^2. Because $n^2 = \Theta(n^2\log^0 n)$, so case 2 selected, k = 0. T(n) = $\Theta(n^2\log n)$
e) c=log2(2)=1, f(n)=$n\log_2 n$. Because $n\log_2 n = \Theta(n^1\log^1 n)$, so case 2 seleccted, k=1. T(n)= $\Theta(n\log^2 n)$

Question 6:

a)

Index	A	B	C	D	E
Value	0	∞	∞	∞	∞

b) After 1 iteration

Index	A	B	C	D	E
Value	0	7	∞	∞	∞

c)

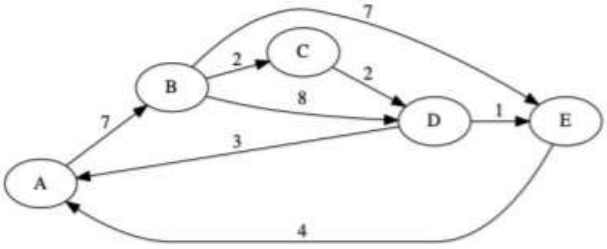
Index	A	B	C	D	E
Value	0	7	9	15	14

d)

Index	A	B	C	D	E
Value	0	7	9	11	14

e)

Index	A	B	C	D	E
Value	0	7	9	11	12



(a) (2 points) What is the initial Distance Array?
(b) (2 points) What is the Distance Array after 1 iteration?
(c) (2 points) What is the Distance Array after 2 iteration?
(d) (2 points) What is the Distance Array after 3 iteration?
(e) (2 points) What is the Distance Array after 4 iteration?

Question 8:

L = [] # declare an empty list as a storage

L.append(RootNode) # Insert the root of the tree to the array

While len(L) > 0: # While list not empty

List L pop the first item or element at index 0

Print the popped item.

L.insert(0,right child of the popped element) #Insert the right child of the popped element to the front of the list.

L.insert(0, left child of the popped element) #Insert the left child of the popped element to the front of the list, the right child will be pushed to the next index.

Question 7:

Import heapq library

Output = [] #To store the sorted output list

Declare an empty min heap with size k

Insert all 1st element in all the arrays into the min heap.

The min heap will auto sort the items while we insert element to it. So the root is the minimum among the 1st element in all the arrays.

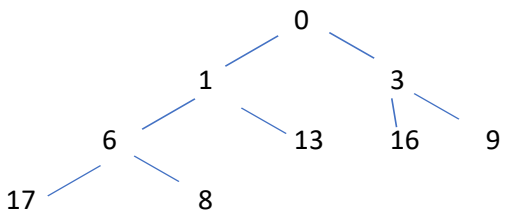
While the min heap is not empty:

Append the root value to the output array

Delete the minimum element from the heap (root)

Insert the next element in the same array of the element deleted. If the next element doesn't exist or array reaches the end, continue.

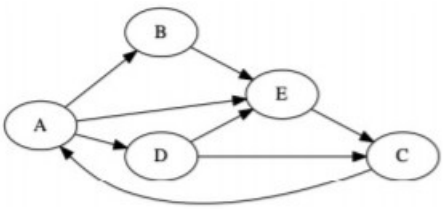
Question 1



Preorder: 0 1 6 17 8 13 3 16 9

Post order: 17 8 6 13 1 16 9 3 0

Question 5



a) Start and end: 1 and 10
b) Start and end: 2 and 7
c) Start and end: 4 and 5
d) Start and end: 8 and 9
e) Start and end: 3 and 6