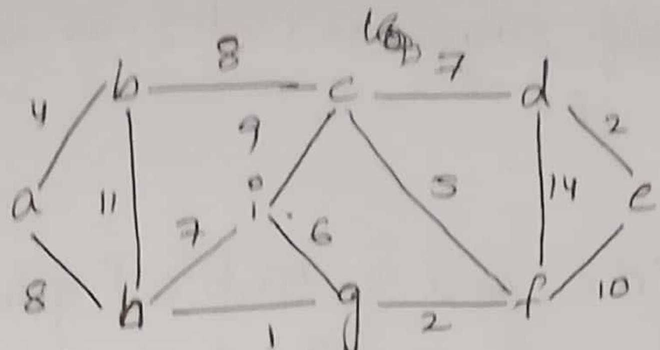
8/7/17

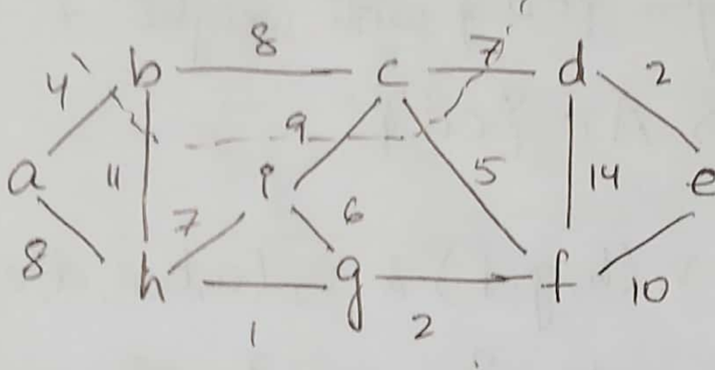
Assignment - 4

1)



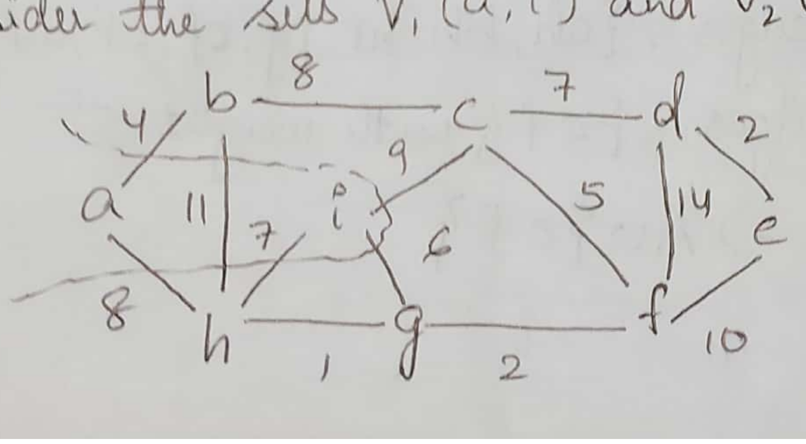
Edges of set A {bc, hg, gf, de}

to find safe edges which are included in set A, we have to cut with respect to set A. Therefore, by cut theorem, making a cut, V, V_1 (b, c) and V_2 (a, h, g, i, f, d, e)



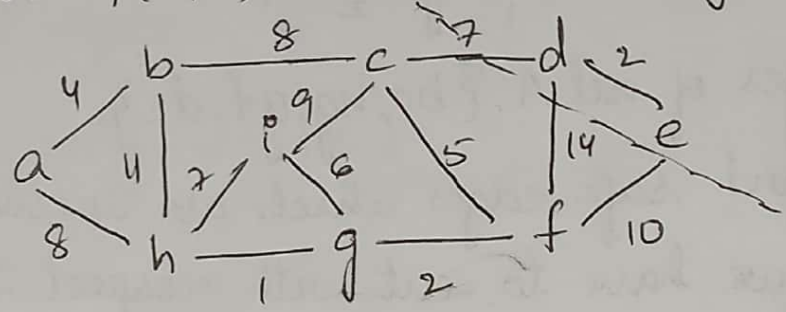
The safe edge out of cross edges, obtained {ab, bh, ci, cf, cd} is {ab} which has a weight of '4'. Therefore set A ∪ {ab}

Consider the sets V_1 (a, i) and V_2 (b, c, d, e, f, g, h)



∴ The cross edges are {ab, bh, ci, hi, ah, ij}
implying the safe edges is {ab} as set A
has {ab} in it.

Consider $V_1(d,e)$ and $V_2(a,b,c,f,g,h,i)$

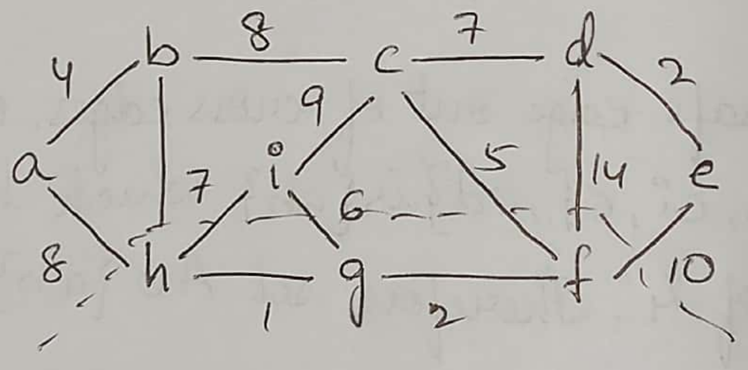


⇒ Cross edges = {cd, df, fe}.

⇒ safe edge = {cd} with weight '7'

⇒ A ∪ {cd}

* Consider $V_1(h,g,f)$ & $V_2(a,b,c,d,e,i)$

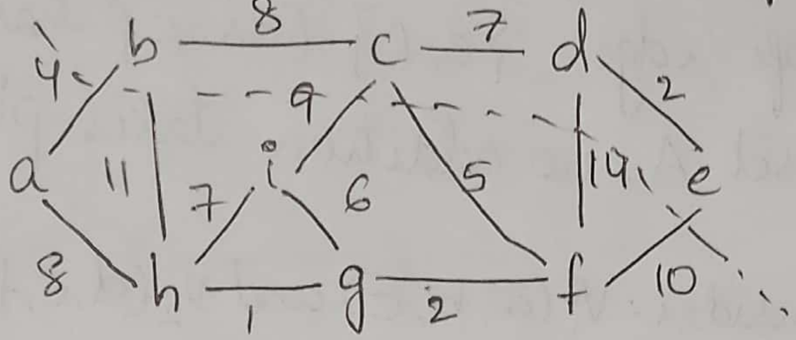


⇒ Cross edges = {ah, bh, hi, ig, cf, ef, df}

⇒ safe edges = {cf} with weight '5'

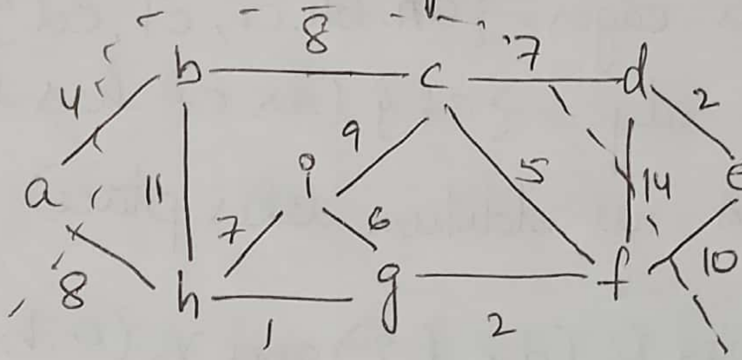
⇒ A ∪ {cf}

* Consider $V_1(b, c, d, e)$ and $V_2(a, f, h, g, i)$



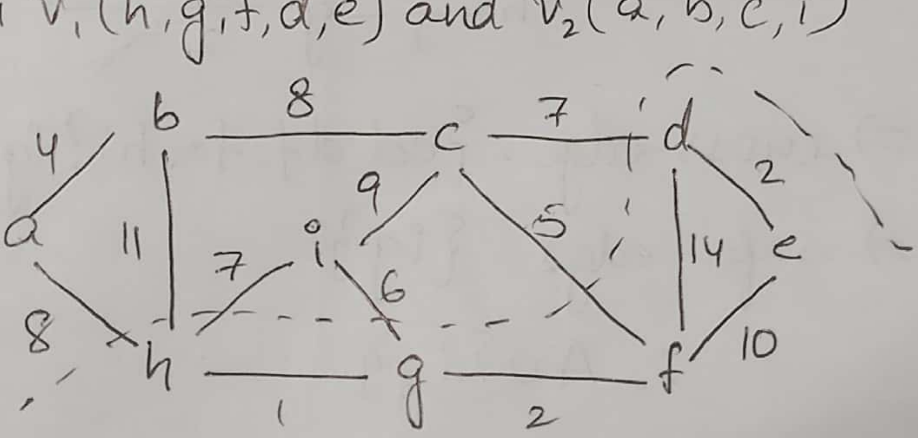
⇒ cross-edges = {ab, bh, ci, cf, fe, df}
⇒ safe-edges = {a, b} (as ab has been added to set A, no addition takes place)

* Consider $V_1(b, c, i, h, g, f)$ and $V_2(d, a, e)$



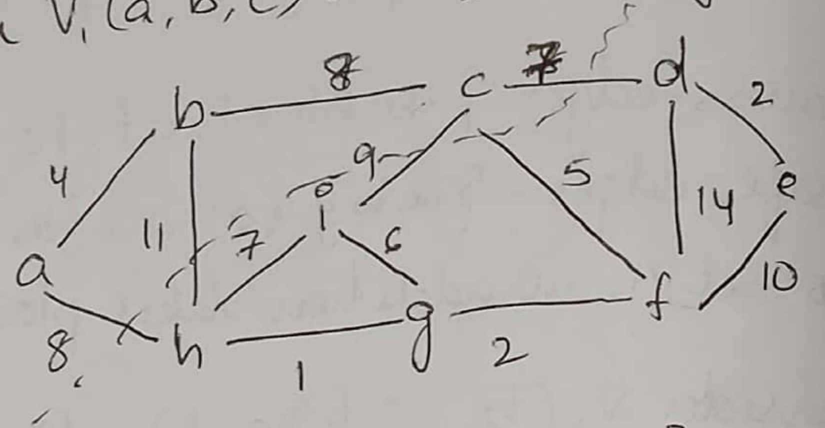
⇒ cross-edges = {ah, ab, cd, df, fe}
⇒ safe-edges = {a, b} (as ab has been added to set A, no additions takes place)

* Consider $V_1(h, g, f, d, e)$ and $V_2(a, b, c, i)$



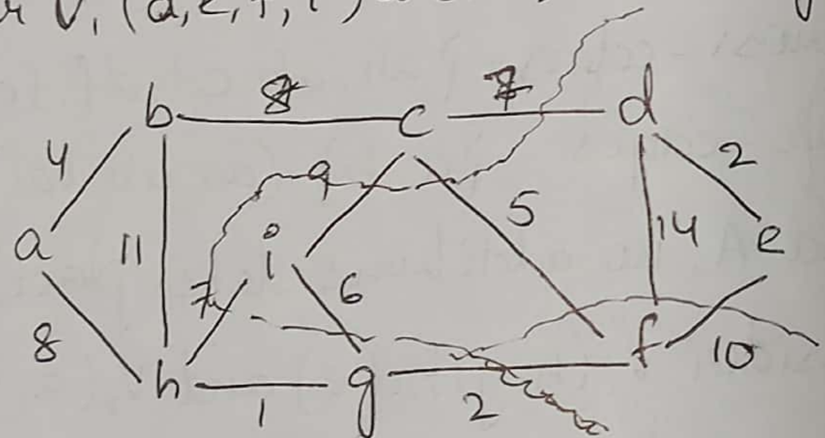
⇒ cross-edges = {ah, bh, ih, ig, cf, cd}
safe-edge = {c, f} (as cf has been added to set A, no addition takes place).

* Consider V_1 (a, b, c) and V_2 (d, e, f, g, h, i)



⇒ cross-edges = {ah, bh, ci, cf, cd}
⇒ safe-edge = {cf} (as cf has been added to set A, no addition takes place)

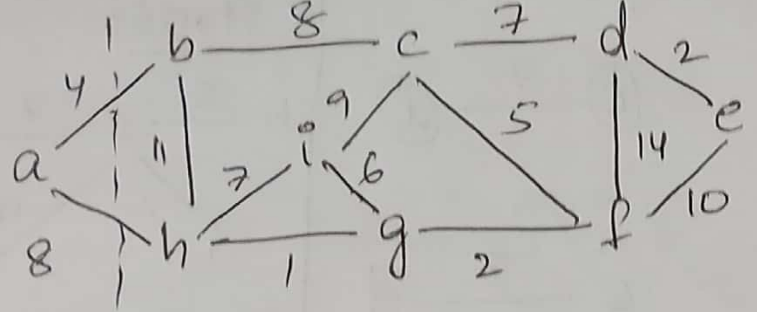
* Consider V_1 (d, e, f, i) and V_2 (a, b, c, g, h, f)



⇒ cross-edge = {cd, df, fe, hi, ig, ci}
⇒ safe-edge = {ig}

∴ A ∪ {ig}

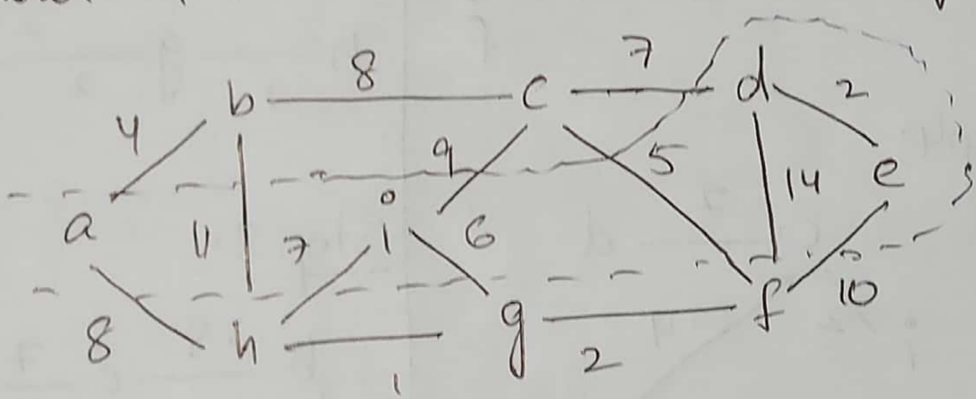
* Consider V_1 (a) and V_2 (b, c, d, e, f, g, h, i)



⇒ cross-edges = {ab, ah}

⇒ safe-edge = {ab} (as ab is already added to set A, no addition takes place)

* Consider V_1 (a, i, d, e) and V_2 (b, c, h, g, f)

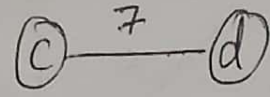


⇒ cross-edges = {ab, ci, cd, fe, df, ig, ih}

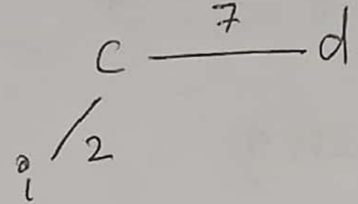
⇒ safe-edges = {ab} (as ab is already added to set A, no addition takes place)

② Prims algorithm:

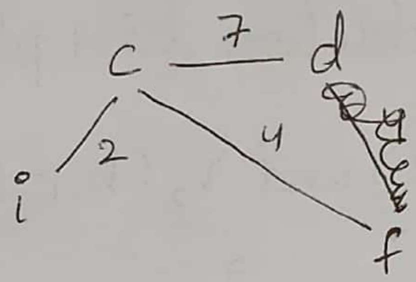
Step 1:



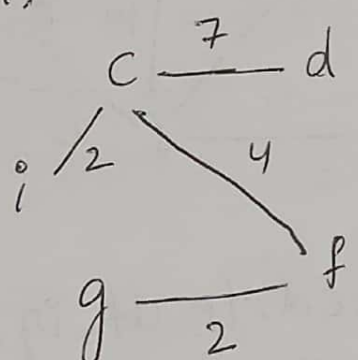
Step 2:



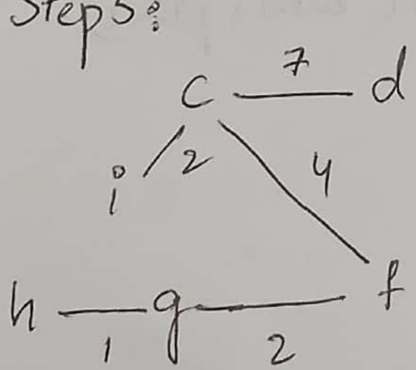
step 3:



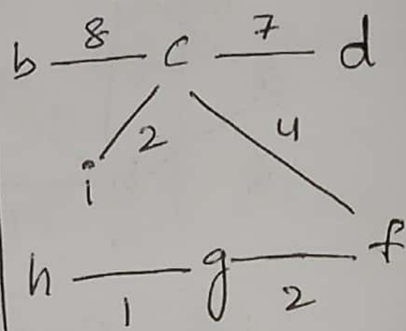
step 4:



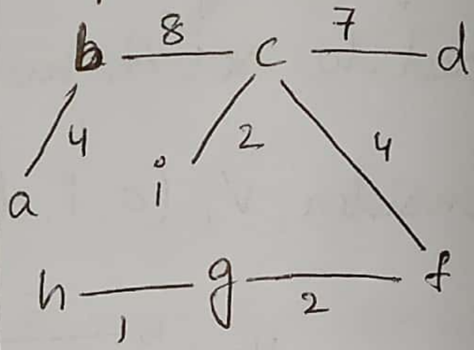
Step 5:



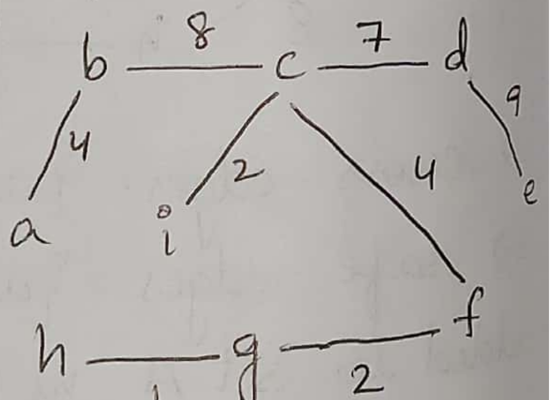
Step 6:



Step 7:



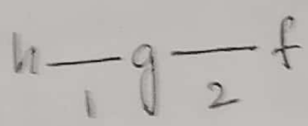
Step 8:



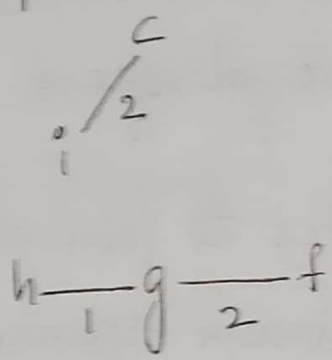
2b) Kruskals algorithm:

Step 1:

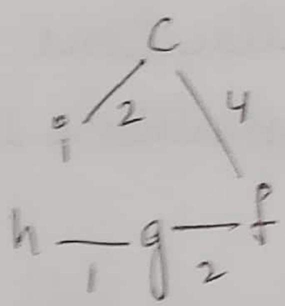
Step 2:



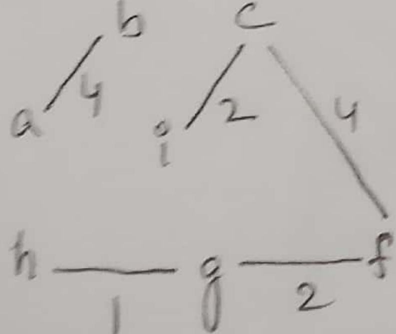
Step 3:



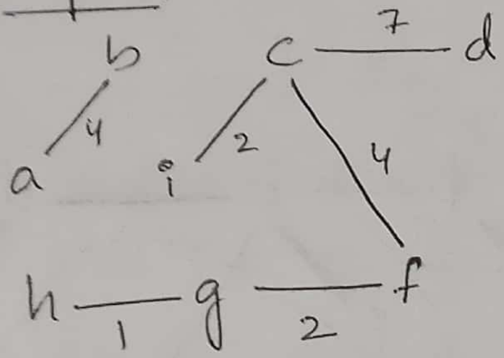
Step 4:



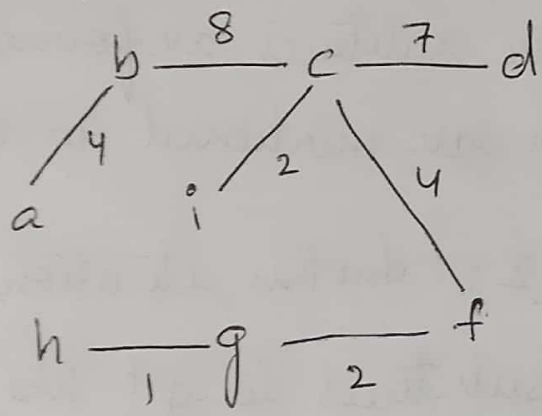
Step 5:



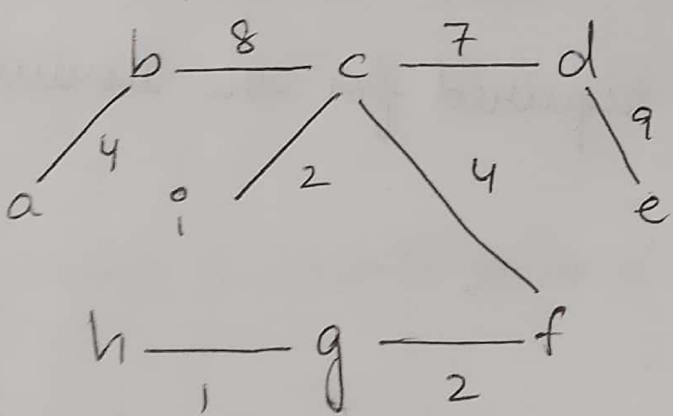
Step-6:



Step-7:



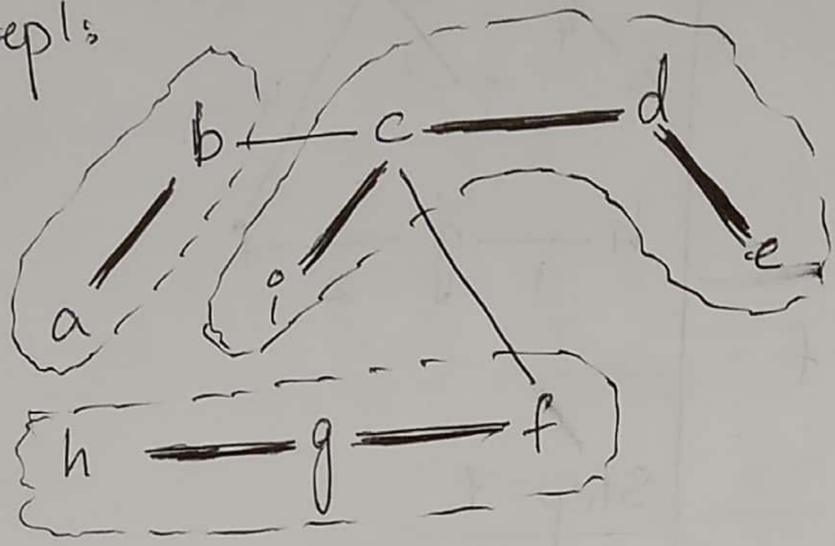
Step-8:



c) Boruvka's algorithm:

* Consider ~~b~~ the cheapest way for every vertex

Step1:



Three subtrees are formed in iteration 1. which are mentioned in above diagram.

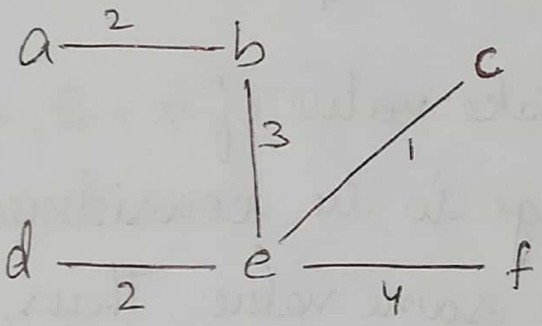
Step 2: Another iteration takes to join all the sub trees to get the minimum spanning tree & MST (T).

Thus, the total number of iterations required for the boruvkas algorithm is 2.

3a) For all values of $x > 3$ the T is still an MST of G.

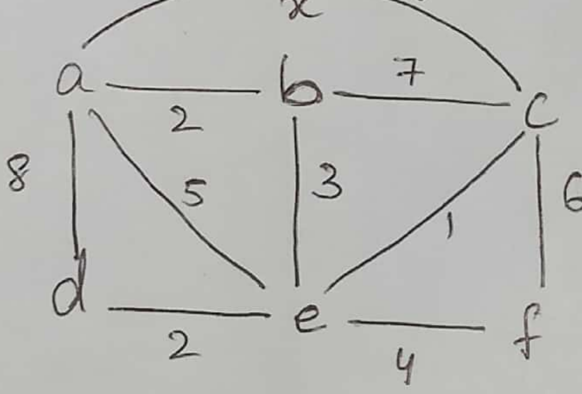
When the values of x is greater than '3', the edge (a,c) will not be considered as the other edges like (c,e) and (a,b) can form a minimum spanning tree by including the vertices a, e into the tree. Thus, the tree T remains intact when the value of $x > 3$.

b) For the given tree T, the MST will be as follows:



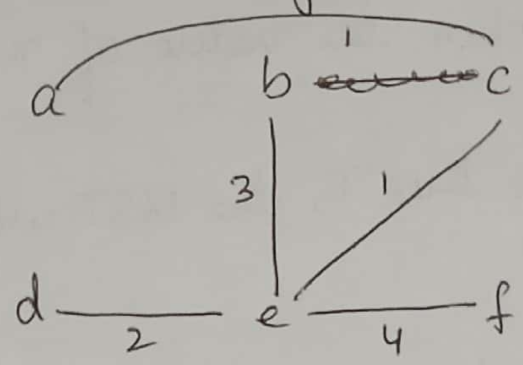
The cost of MST(T) = 2 + 2 + 3 + 1 + 4
= 12.

Now consider, a new edge (a,c) with value 'x'.



The condition is to include the edge (a, c) with value 'x'. So, to consider (a, c) 'x' should be less than values of edges connected to 'a' and 'c'. The least value from edge 'a' is '2' and edge 'c' is 1.

Let us suppose, that value of $x = 1$, the MST will change to new MST as



Similarly we take value of $x = 2$, as there is a chance of edge to be considered as {ab, ac} are of same value. Thus, the values of x can be 1 ~~and~~ 2.