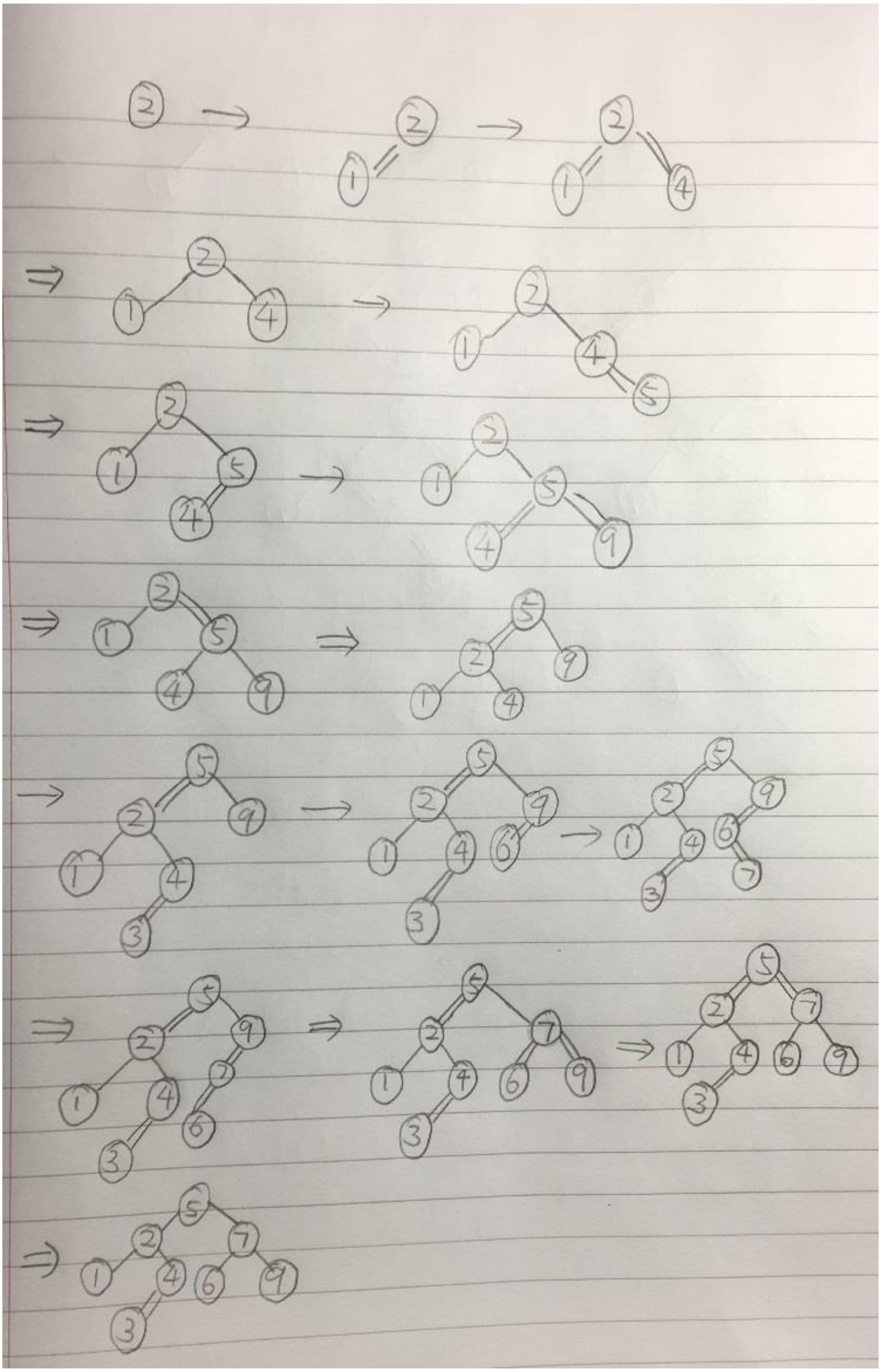
1.

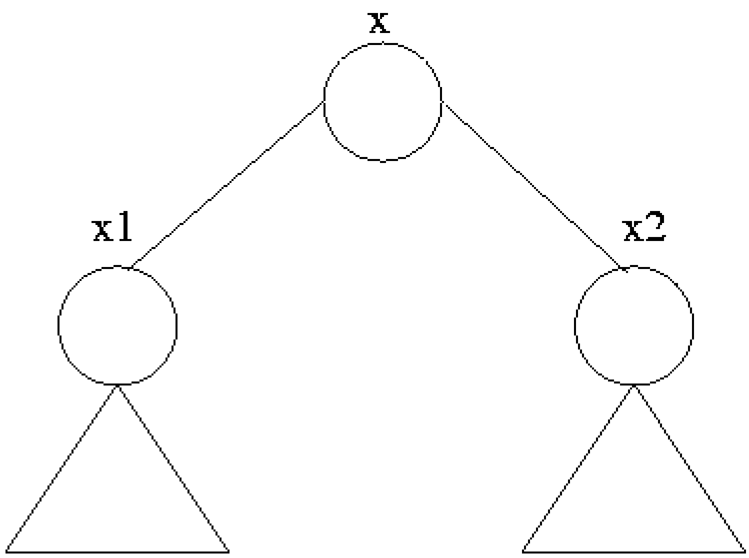


2.

We do this by induction on the height of x. If h(x) = 0, bh(x) = 0, x is leaf and hence the subtree has no internal nodes, as corroborated by

$$2^0 - 1 = 0$$

Let h(x) > 0 and let x1 and x2 be its two children



Note that h(x1), h(x2) are both <= h(x) - 1. Assume the result to be true for x1 and x2. We shall show the result is true for x.

Now,

$$bh(x_1) \le bh(x) \quad \text{and} \quad \ge bh(x) - 1$$
$$bh(x_2) \le bh(x) \quad \text{and} \quad \ge bh(x) - 1$$

Therefore, the tree with root x1 has at least

$2^{bh(x) - 1} - 1$ internal nodes

whereas the tree with root x2 has at least

$2^{bh(x) - 1} - 1$internal nodes

Thus the tree with root x has at least

$1 + 2^{bh(x) - 1} - 1 + 2^{bh(x) - 1} - 1 = 2^{bh(x)} - 1$; internal nodes

let h be the height of the tree. Then

bh (root) >=h/2

Thus

N is number of internal nodes

$$N + 1 \geq 2^{h/2}$$

$$\log(N + 1) \geq h/2 \Rightarrow$$

$$h \leq 2 \log(N + 1)$$

3.

Since this algorithm dose not removal edges that would disconnect the graph. Then it produce a tree.

In class we proved that kruskal algorithm produce a MST by contradiction.

We can do the same. Suppose T' is the real MST and F is the MST made by Reverse-Delete algorithm.

Every edges have same weight except 'e' and 'f'.

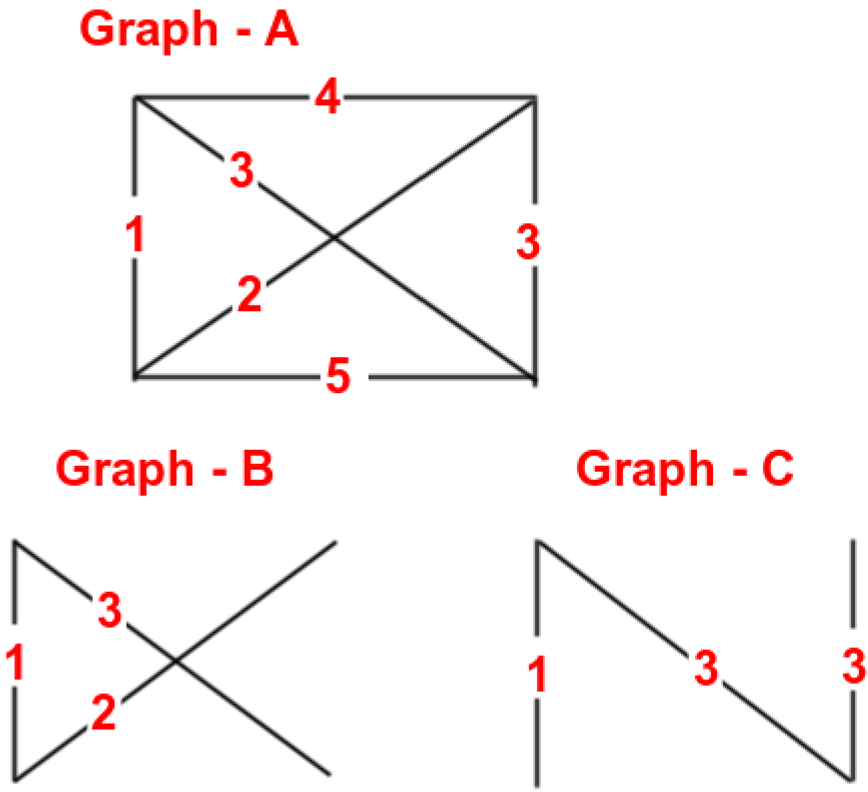
wt(f) < wt(e) this is impossible since this causes the weight of tree T' to be strictly less than T . since T is the minimum spanning tree , this is simply impossible.

wt(f) > wt(e) this is also impossible. since then when we are going through edges in decreasing order of edge weights we must see " f " first . since we have a cycle C so removing " f " would not cause any disconnectedness in the F. so the algorithm would have removed it from F earlier . so " f " does not exist in F which is impossible.

4.

Suppose there are two minimum trees, AA and BB. Let ee be the edge in just one of A,BA,B with the smallest cost. Suppose it is in AA but not BB. Suppose ee is the edge PQPQ. Then BB must contain a path from PP to QQ which is not simply the edge ee. So if we add ee to BB, then we get a cycle. If all the other edges in the cycle were in AA, then AA would contain a cycle, which it cannot. So the cycle must contain an edge ff not in AA. Hence, by the definition of ee (and the fact that all edge-costs are different) the cost of ff must be greater than the cost of ee. So if we replace ff by ee we get a spanning tree with smaller total cost. Contradiction

5.



B is MST and C is MBST