Homework Assignment 6
CS 430 Introduction to Algorithms

Name : Karthik Mahesh

CWID : A20383027

1.Solution:

This algorithm is very similar to the binary Huffman algorithm mentioned in the CLRS.
In this algorithm, instead of taking minimum 2, we select minimum 3. But since the number of symbols may not form complete 3-ary tree, we add some dummy variable with value=0 (highest priority).

```
Huffman_Ternary(C)
{
  if |C| = even
  then add a dummy character z with frequency 0.

  N = |C|
  Q = C  //heapifying the characters

  for i = 1 to floor(N/2)
  {
    allocate new_node
    left[new_node] = U = extract_min(Q)
    mid[new_node] = V = extract_min(Q)
    right[new_node] = W = extract_min(Q)
    F[new_node] = F[U] + F[V] + F[W];
    insert(Q , new_node);
  }
  return extract_min(Q);
 }
```

Proof:
Any optimal tree has the lowest three frequencies at the lowest level.
Proof by contradiction:
Suppose it is not the case then we could switch a leaf with a higher frequency from the lowest level with one of the lowest three leaves and obtain a lower average length. Without any loss of generality, we can assume that all the three lowest frequencies are the children of the same node. If they are at the same level, the average length does not change irrespective of where the frequencies are. They only differ in the last digit of their code word (one will be 0,1 or 2).
Again as the binary code words we have to contract the three nodes and make a new character out of it having ‘frequency = total of three node's(character's) frequencies’. Like binary Huffman codes, we see that the cost of the optimal tree is the sum of the tree with the three symbols contracted and the eliminated sub-tree which had the nodes before contraction. Since it has been proved that the sub-

tree has to be present in the final optimal tree, we can optimize on the tree with the contracted newly created node.

2.Solution:
a.

Form an increasing ordered list by sorting all the jobs according to the execution time.
Start executing the jobs according to the sorted list.

This schedule will have the minimum total weight time.

3.Solution:

When an edge that represents a communication is removed from operation, we will get 2 sets of vertices S1 and S2.
Now we can use union find algorithm and find the new MST.
Union(S1,S2) -> S
Find(x) -> S : report set containing x

Time complexity : O(log n)

4.Solution:

Yes. The MST remains the same after adding same amount of weight to every edge to make all the edges positive because when all the edges are increased equally, the ratio between the edges remain same. So the MST also remains the same.

References:
Analysis and Design of algorithms textbook
https://stackoverflow.com