Homework 7: Q2

Name: Xinkai Lin, xinkaili

*Don't forget to input your list of collaborators and sources on **AutoLab**.*
Please submit this file as a PDF.

1 Algorithm Idea

Since the bounding communication cost for this algorithm will be O(MST(G) * M). To solve this problem, first I will be using Reverse-Delete-Algorithm to find out the Minimum Spanning Tree for the graph G. Then, assume t as its root node. For all A_i exist in MST, from leaf node to root node, compute A_i with A_{i+1} in the form of binary representation base on M. If $A_i[j]$ and $A_{i+1}[j]$ = 1, then B[j] = 1, else B[j] = 0. Finally, return B.

2 Algorithm Details

```
Input: G= (V, E), c_e > 0 for every e in E      // see Reverse-Delete-Algorithm on lecture MST
T = E
Sort edges in decreasing order of their cost
Consider edges in sorted order
    If an edge can be removed T without disconnecting T then remove it
q = {}   //queue
t = root
For all node in T   1 ≤ i ≤ n'
    If i is leaf node do
        Add i to the q
    End if
End for
While q is not empty do
   Current = q.pop ()
   Compute current with its parent node   //this will compute the current node with its parent
                                          //node to find their intersection in binary representation
   If parent node has not been visited and not equal to t do
        Add parent node to q
   End if
End while
Return B        // which contains t's binary representation
```

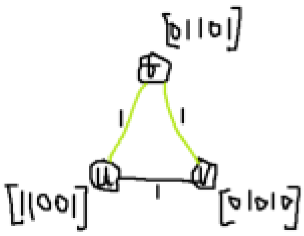
3 Proof of Correctness Idea

To prove the algorithm's correctness, we will show that the intersection of all A and it may be empty. First of all, I use the Reverse-Delete-Algorithm to find out the Minimum Spanning Tree.

Then, from there I will compute all the node exists in MST from the leaf node to the root in the form of binary representation base on the M. For instance, n=3, m=5. Where we have {t, u, w: (t, w), (t, u), (u, w). t is the root node. The MST for this will be (t, w) and (t, u). Then, I will compute both w with t and u with u. After the algorithm end, it will return an array which contains the intersection of all nodes. If they are intersected then it will be 1, else 0. So, this algorithm is correct.

4 Proof Details

In this detail, I will use the Example 1 from the recitation note to demonstrate this problem.



In this example. In the graph an MST is shown with the green edges. Which is (t, u) and (t, v). Now, we compute them with their binary representation. Since t and u are intersected on 2 and 5, then we can write t as [0, 1, 0, 0, 1], remember that if they intersected we assigned it to be 1, else 0. Then, we compute t with v. [0, 1, 0, 0, 1] and [0, 1, 0, 1, 0] has only one intersected point which is 2, so we rewrite t as [0, 1, 0, 0, 0]. This is our result, which is B. With the same algorithm, we can always find the output by finding the MSG first and then compute them with any giving n and m inputs.

5 Bounding Communication Cost

For this algorithm, the communication cost will be O(MST(G) * M) where M will be giving, and I will use Reverse-Delete-Algorithm to find the MST(G). For all edge $e_i \in E$ in MST,
$(e_i + e_{i+1} + \ldots + e_{n'}) * M$.