CSCI 405

Trevor Rivers

November 2021

1

1.1 Give give an algorithm with running time O(m + n) that determines whether the m judgements are consistent.

```
augmentedBFS(s)
    for(u ∈ V - {s})
        u.label = 0
    s.label = 1
    Q = {}
    enque(Q,S)
    while(Q != {}){
        u = deque(Q)
        for(v ∈ G.Adjacency[u]){
            if(v.label == 0)
                if(judgement(u,v) = 'same')
                    v.label = u.label
                else
                    v.label = - u.label
                enque(Q,v)
        }
    }
    ∀ edges(u,v){
        if(judgement(u,v) = 'same')
            if(v.label != u.label)
                return false
        else
            if(v.label == u.label)
                return false
    }
    return true
```

Explanation:

This algorithm stems from breadth first search where it's under the assumption that the nodes of a graph are connected. When the graph isn't connected, the algorithm runs for every other part that is connected. This begins by choosing some arbitrary node 's' and labeling it "1". This is a representation that it is indeed the starting node. When breadth first search is active, some node is visited through the edge(u,v). This outputs the label 'u' as well as the judgement for the the edge (u,v) (this results in 'v' being labeled). When 'u' is labeled "1" and the judgement for the edge (u,v) is assigned 'same', 'v' is labeled as 1. However, this implies that the judgement for the edge (u,v) is 'different'. This results in 'v' being labeled "-1". The procedure makes it so that each and every node has a custom label from the tree edges where some arbitrary node 's' is labeled "1". After this, it goes through the non - tree edges for (u,v) and checks to see if the endpoint (u,v) is consistent or not with the judgement for the edge. The output will return true if and only if the correlation of judgments are consistent between the endpoints (u,v)(from the non - tree edges), and for the edge. Otherwise, it will return false.

If the augmentedBFS algorithm returns false, this implies that that the judgements were inconsistent with the endpoints and the edge. This is the case because like I mentioned before, the nodes had a custom label where they were swapped between the label of "-1" and "1". What this means, is that the following labeling was inconsistent for some non-tree edge. With that in mind, I know that there isn't the possibility to change the labels in order to make it so that the edges will be consistent. If the augmentedBFS algorithm returns true, I know that the judgements are consistent.This is the case as the algorithm labeled all available nodes in the tree using the labels of "-1" and "1". This procedure makes sure that the labels are consistent with the tree edges. The algorithm makes sure to check that the labels are consistent with the non tree edges. Since the algorithm is an augmented version of breadth first search (with other minor time complexity adjustments),this will will result in the running time of O(m + n). It takes O(n) amount of time to execute some conditions internally and O(m) amount of time to scan the non - tree edges when the augmented breadth first search is completed.

1.2 What is the minimum number of judgements m that they must make...?

Explanation:

The minimum number of judgements needed are n - 1 for an n sized spanning tree. From prior knowledge, I know that they don't have any cycles and they are already connected minimally. Further,there are no disjointed nodes and it was stated that there were no mistakes made (which would make a difference if there were).When traversing an n sized tree, it's being traversed one node at a time. Thus resulting in n - 1.