Saaif Ahmed

Wednesday, April 15, 2020 7:09 PM

Lab 6

Define the subproblem V(u): If u is a vertex that is not taken into the vertex cover then the other vertices surrounding it must be taken. This means the either the child or the root is taken. The better of the two is taken.

The algorithm is as follows:

```
    Create a table that will hold solutions for a given node for the subproblem V(u) as defined above.
    Start from a leaf node.
    Run the method V(u) on the node.
        //What this does is it takes the node and removes all nodes connected to it
        //From there is takes the remaining nodes are runs V() on them, it returns a set of nodes that
        solve the Vertex cover when ran on a node.
    Store the solution of V(u) in the table
    Run the method V(u) on the nodes in the previous indices of the table
        //this should be constant as it is pulled from the table except for the leaf nodes
    Compare the results, if the V(u) at u is better than at the previous table indexes put node u at the
    table slot
        Otherwise take the children and put it into the table slot.
    Repeat for all nodes in the tree structure.
    Return the value in the table[V][V]
```

Analysis:

This algorithm is linear. We run the algorithm for all the nodes V in the graph and each time check E amount of edges. There are cases where there are repeats such as two leaf nodes connected to the same parent. Because we store the answers in a table, the lookup from it is constant O(1).
Thus algorithm is at most O(V *2E) or O(V*E).