I collaborated with: W3029343

Problem
Answer Question 11 from Chapter Three of your text *Algorithm Design*. In addition to designing the algorithm, justify its correctness and time and space requirements.
Solution

Algorithm 1 determines whether a virus introduced at computer a at time x could have infected computer b by time y

```
function DETECTION(C_a, x, C_b, y, m)
    let G be an empty directed graph
    for all triplets (C_i, C_j, t_k) in m do
        i ← node (C_i, t_k)
        j ← node (C_j, t_k)
        add {i, j} and {j, i} to G
        if we have seen C_i or C_j before then
            add {(C_i, t_o), (C_i, t_k)} and/or {(C_j, t_o), (C_j, t_k)} to G where o is the previous time
        end if
    end for
    Travel to neighboring node reachable from C_a, t where t ≥ x
    run BFS from this node
    if we reach a node of the form (C_b, t) where t ≤ y then
        return True
    end if
    return False
end function
```

In this algortithm, we build a graph G that keeps track of the flow of the virus since every edge in G points forward in time. We start looking through the graph once C_a connects to another computer C_i. We also keep track of the computers C_i interacts with and which computers those interact with. Eventually, a computer that has an edge between a C_j, t_j and C_b, t_j where $x \leq t_j \leq y$. Thus, we will see if C_b gets infected.
Time requirement: Adding a node to G: $O(m)$ + BFS: $O(m)$ = $O(2m)$.
Space requirement: G: $O(m)$