Deanna Omron
Shen
CS-210
November 1, 2022

Part B - Time Complexity of Program 3

Pseudocode -

```
Find levels of e1 & e2;
        If (e1 && e2's ptrs do not exist) {
                return null ptr;
        }
        If (e1 && e2 ptrs match) {
                return parent prt;
         } else if (e1's level > e2's level) {
                decrement e1's level;
                if (e1 && e2's parent ptrs match) {
                        return parent ptrs;
                } else if (e1 is e2's parent ptr){
                        return e1's ptr;
                } else if (e2 is e1's parent ptr) {
                        return e2's ptr;
                }
        } else if (e2's level > e1's level) {
                decrement e2's level;
                if (e1 && e2's parent ptrs match){
                        return parent ptrs;
                } else if (e2 is e1's parent ptr){
                        return e2's ptr;
                } else if (e1 is e2's parent ptr){
                        return e1's ptr;
                }
        } else if (e1 level = e2 level) {
                if (e1 && e2's parent ptrs match){
                        return parent ptrs;
                } else{
                        decrement e2 & e1's levels by one;
                        if (e1 && e2's parent ptrs match) {
                        return parent ptrs;
                        }
                }
}
```

Explanation

The worst case scenario for findClosestSharedManager would be if one employee is present and the other is not. This would present a time complexity of O(H). This is because we start with the employee that has the lowest level (which can be at most a height of h) and work our way up to see if both employees point to the same parent node. At worst one the algorithm would be checking every step from level H to 0 to see if two pointers are pointing to the same parent node. Each of these comparisons takes O(1) time complexity. Due to starting at the bottom most level of one of the employees and working our way up through comparisons from a height of H, this function has a time complexity of O(height of the tree) or O(H).