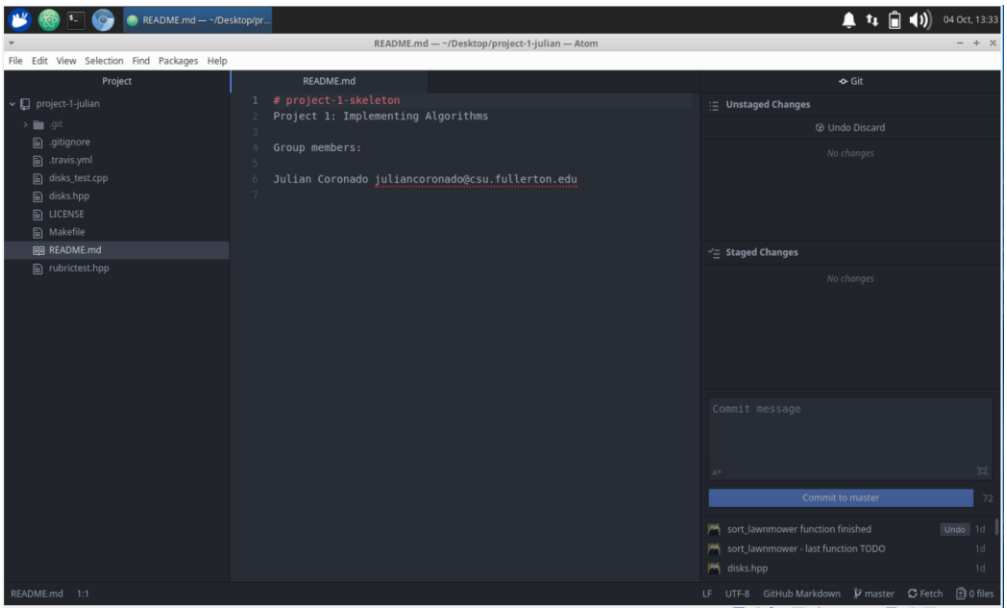
Julian Coronado
juliancoronado@csu.fullerton.edu
335 Project 1

Implementing Algorithms



Psuedocode

Left to Right Algorithm

```
sort_left_to_right(disks) {

    swaps = 0;
    while (disks is not sorted) {
        for (i from 0 to disk count - 1) {
            if (disks at index i is DISK_DARK and disks at index i + 1 is DISK_LIGHT) {
                swap(disks at index i with the next index)
                swaps++
            }
        }
    }

    return sorted_disks(disks)
}
```

Lawnmower Algorithm

```
sorted_disks sort_lawnmower(disks) {
    swaps = 0;
    while (disks is not sorted) {

        // sorting goes to the right
        for (i from 0 to disk len - 1) {
            if (disk at index i is DISK_DARK and disks at index i + 1 is DISK_LIGHT) {
                swap(disk at index i with the next index)
                swaps++
            }
        }

        // sorting goes back to the left
        for (j from disk len - 1 to 0, decrement by 1) {
            if (disk at index j is DISK_LIGHT and disk at index j - 1 is DISK_DARK) {
                swap(disk at index j - 1 with disk at index j)
                swaps++
            }
        }
    }
    return sorted_disks(disks, swaps)
}
```

Julian Coronado
juliancoronado@csu.fullerton.edu
335 Project 1

Time Complexity for Algorithms

```
sort_left_to_right(disks) {

    swaps = 0; 1
    while (disks is not sorted) {    n / 2
        for (i from 0 to disk count - 1) { n
            if (disks at index i is DISK_DARK and disks at index i + 1 is DISK_LIGHT) { 1
                swap(disks at index i with the next index) 1
                swaps++ 1
            }
        }
    }

    return sorted_disks(disks) 1
}
```

$T(n) = 1 + (n/2)(n(1+1+1))$

$T(n) = 1 + (3/2)n^2 \in \mathbf{O(n^2)}$

```
sorted_disks sort_lawnmower(disks) {
    swaps = 0; 1
    while (disks is not sorted) {  n / 2

        // sorting goes to the right
        for (i from 0 to disk len - 1) { n
            if (disk at index i is DISK_DARK and disks at index i + 1 is DISK_LIGHT) { 1
                swap(disk at index i with the next index) 1
                swaps++ 1
            }
        }

        // sorting goes back to the left
        for (j from disk len - 1 to 0, decrement by 1) { n
            if (disk at index j is DISK_LIGHT and disk at index j - 1 is DISK_DARK) {  1
                swap(disk at index j - 1 with disk at index j) 1
                swaps++ 1
            }
        }
    }
    return sorted_disks(disks, swaps) 1
}
```

$T(n) = 1 + (n/2)(n(1+1+1) + n(1+1+1))$

$T(n) = 1 + (3)n^2 \in \mathbf{O(n^2)}$