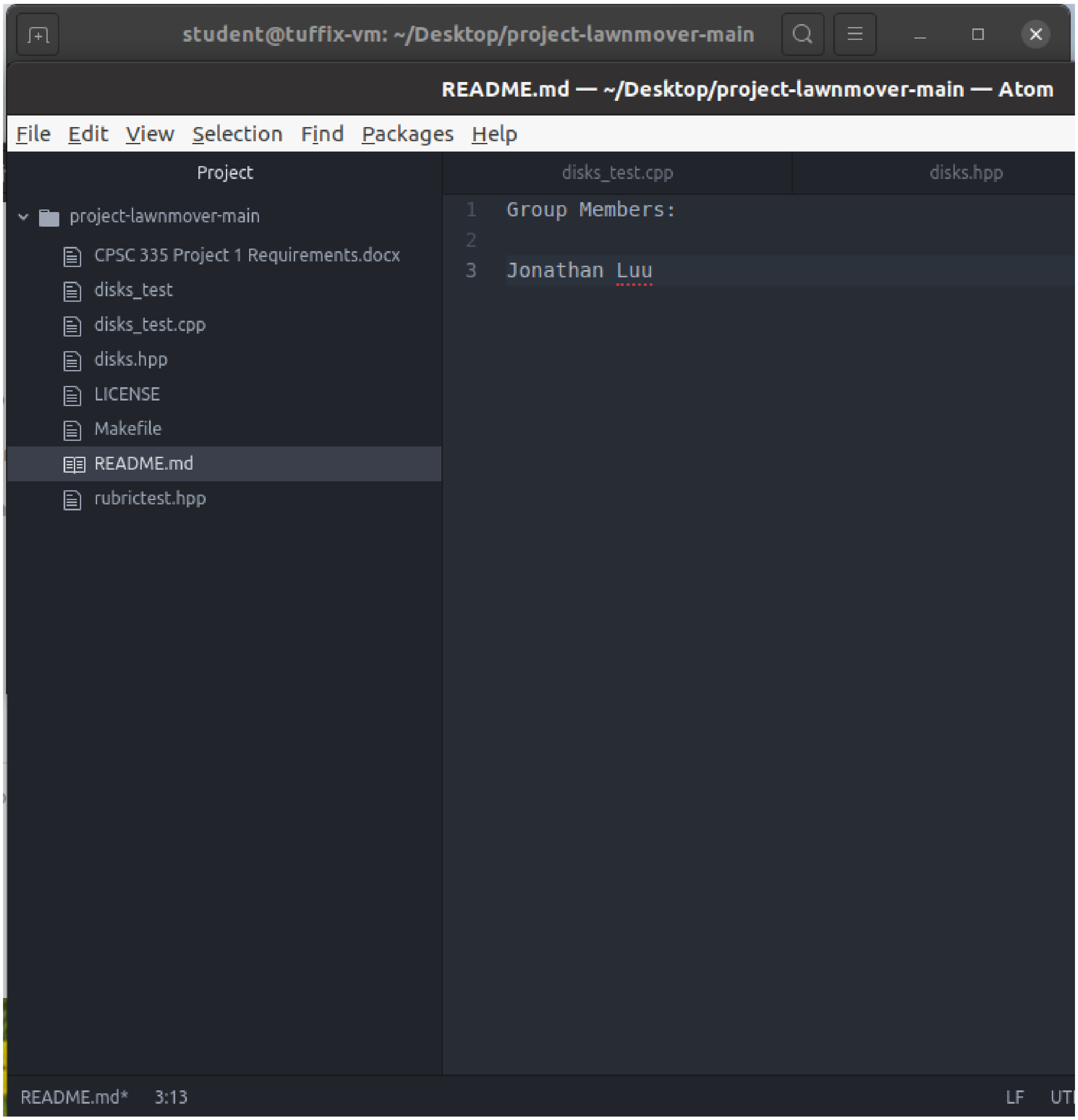
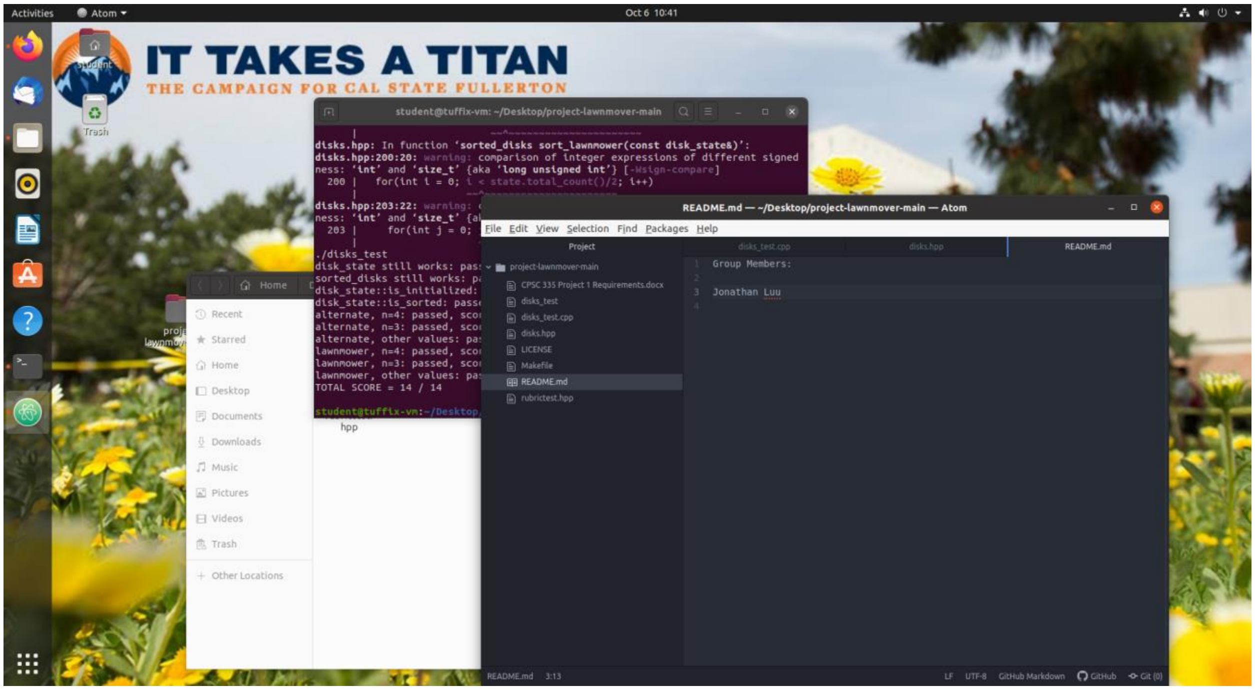
Jonathan Luu

jqluu@csu.fullerton.edu

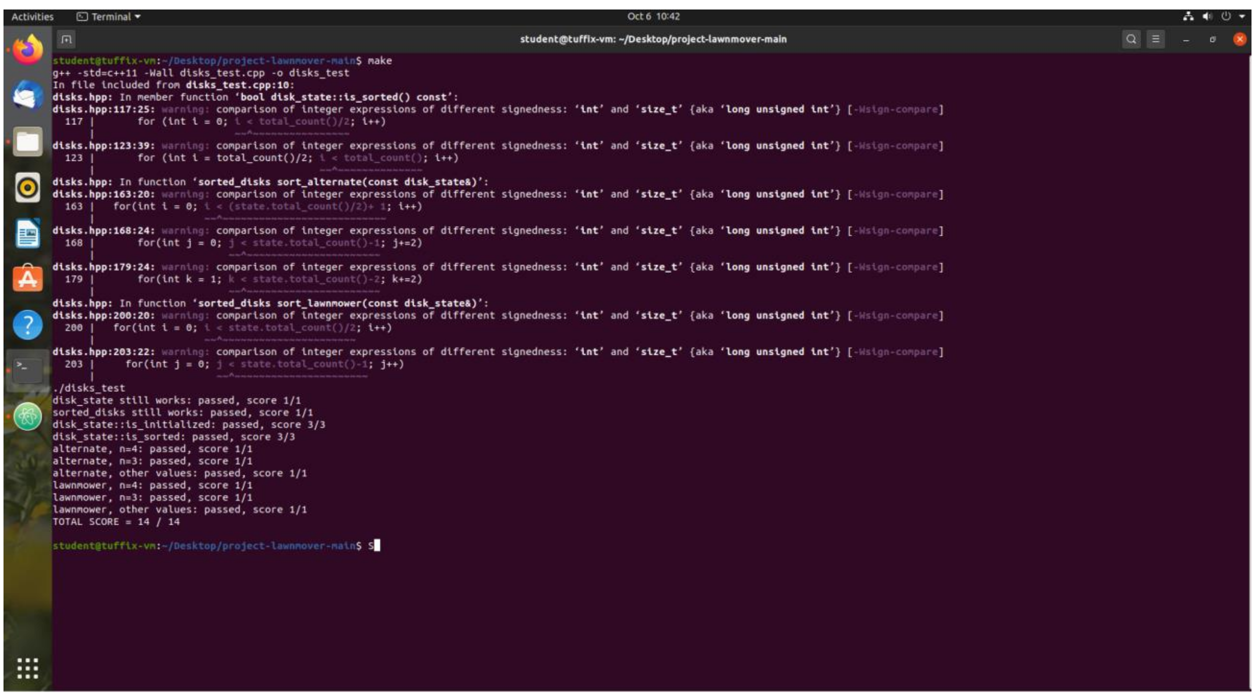
Project 1

2. Tuffix/Atom screenshot






3. Compiling/executing



```
./disks_test
disk_state still works: passed, score 1/1
sorted_disks still works: passed, score 1/1
disk_state::is_initialized: passed, score 3/3
disk_state::is_sorted: passed, score 3/3
alternate, n=4: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 14 / 14

student@tuffix-vm:~/Desktop/project-lawnmover-main$
```

4. Pseudocode

```
def lawnmower (int n, int a[])
{                                                         Steps

int numSwaps = 0;                                           1

    for(n/2 times)                                          n/2
    {
        current = 0;                                        1

        for (2n-1 times)                                    2n-1
        {
            if (a[i] == black && a[i+1] == white)           6
            {
                current = a[i]                              2
                a[i] = a[i+1]                               4
                a[i+1] = current                            3

                numSwaps++                                  1
            }
        }

        for (2n-1 times)                                    2n-1
        {
            if (a[i] == white && a[i-1] == black)           6
            {
                current = a[i]                              2
                a[i] = a[i-1]                               4
                a[i-1] = current                            3

                numSwaps++                                  1
            }
        }
    }

return a[] and numSwaps                                     1
}

STEP COUNT:
1 + n/2 * (1 + 2n-1 * (6 + 2 + 4 + 3 + 1) + 2n-1 * (6 + 2 + 4 + 3 + 1)) + 1
= 2 + n/2 * (1 + 2n-1 * (16) + 2n-1 * (16))
= 2 + n/2 * (64n - 31)

Step count = 32n^2 - 31/2n + 2
```

```
def alternate(int a[], int n)
{                                                           Steps
    int numSwaps = 0                                          1

    for(n+1 times) // 2n is length                           n+1
    {

        if (curr % 2)                                         1
        {
            for (i = 0, i < 2n-1, i+2)                        n
            {
                if (a[i] == black && a[i+1] == white)         6
                {
                    current = a[i]                            2
                    a[i] = a[i+1]                             4
                    a[i+1] = current                          3

                    numSwaps++                                1
                }
            }
        }

        else    //odd
        {
            for (i = 1, i < 2n-1, i+2)                       n-1
            {
                if (a[i] == black && a[i+1] == white)         6
                {
                    current = a[i]                            2
                    a[i] = a[i+1]                             4
                    a[i+1] = current                          3

                    numSwaps++                                1
                }
            }
        }

    }

    return a[] and numSwaps                                   1
}

Step Count:
1 + n+1 * (1 + max(n+6+2+4+3+1, n-1+6+2+4+3+2+1)) + 1
= 2 + n+1 * (n + 17)
= 2 + n^2 + n + 17 + 17n

Step count = n^2 + 18n + 19
```

5. Time complexity proof for algorithms

a. Lawnmower-

The step count for the lawn mower algorithm is  $32n^2 – 16n + 2$,

(16 is simplified from 31/2)

We want to prove that it is **O(n^2)**
Assuming that n > 1,

$$\frac{f(n)}{g(n)} = \frac{32n^2 \ - \ 16n \ + \ 2}{n^2} < \frac{32n^2 - 16n^2 + 2n^2}{n^2} = 18$$

C = 18

Therefore,
$32n^2 – 16n + 2$ is **O(n^2)** because
$32n^2 – 16n + 2 <= 18n^2$, whenever n > 1

b. Alternate-

The step count for the alternate algorithm is  $n^2 + 18n + 19$

We want to prove that it is **O(n^2)**
Assuming that n > 1,

$$\frac{f(n)}{g(n)} = \frac{n^2 + 18n \ + 19}{n^2} < \frac{n^2 + 18n^2 \ + 19n^2}{n^2} = 38$$

C = 38

Therefore,
$n^2 + 18n + 19$ is **O(n^2)** because
$n^2 + 18n + 19 <= 38n^2$, whenever n > 1