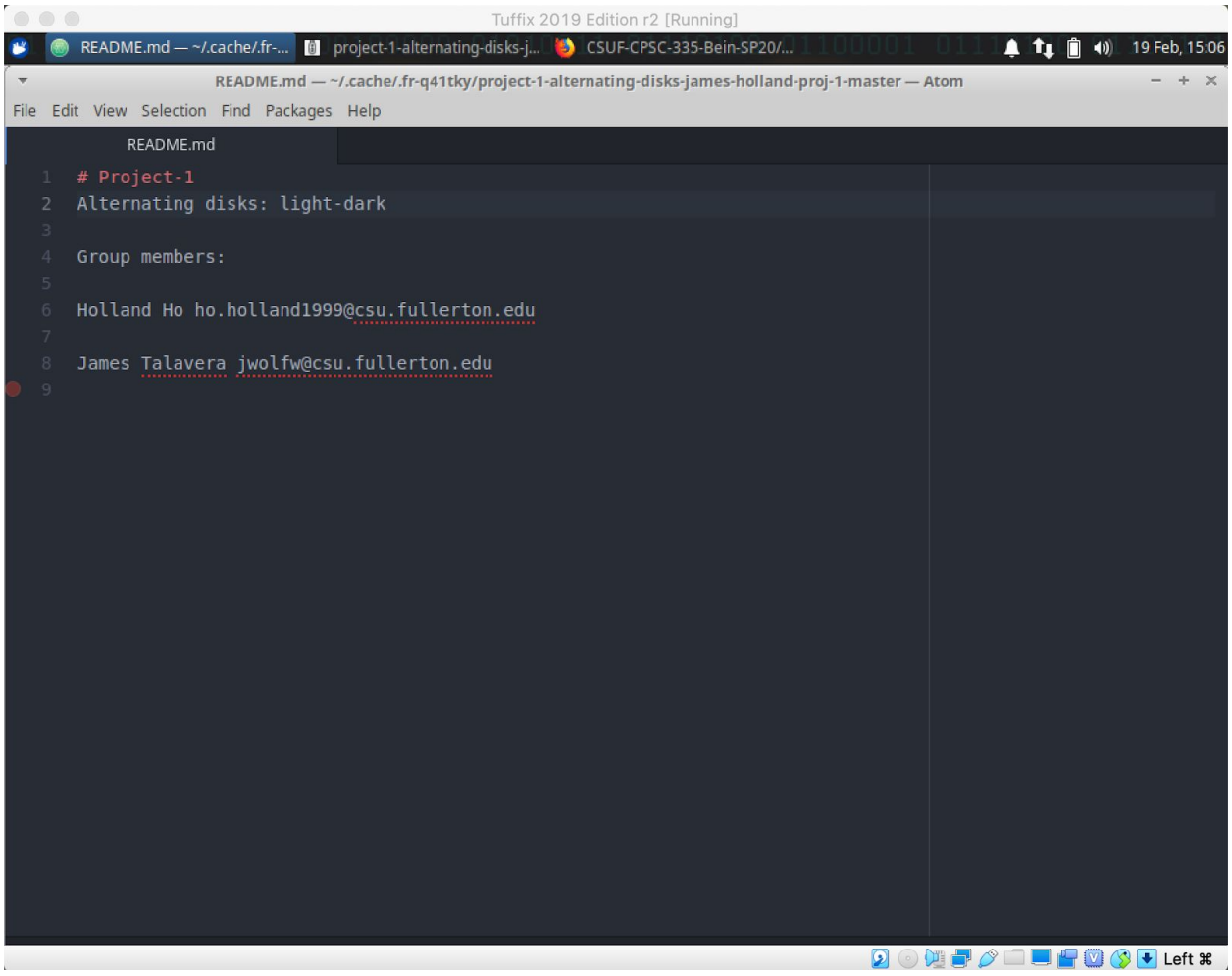
Project 1 Brief Written Report

Group Member Names:

Holland Ho ho.holland1999@csu.fullerton.edu

James Talavera jwolfw@csu.fullerton.edu

Screenshot:



Pseudocode listings:

1. **sort_left_to_right algorithm:**

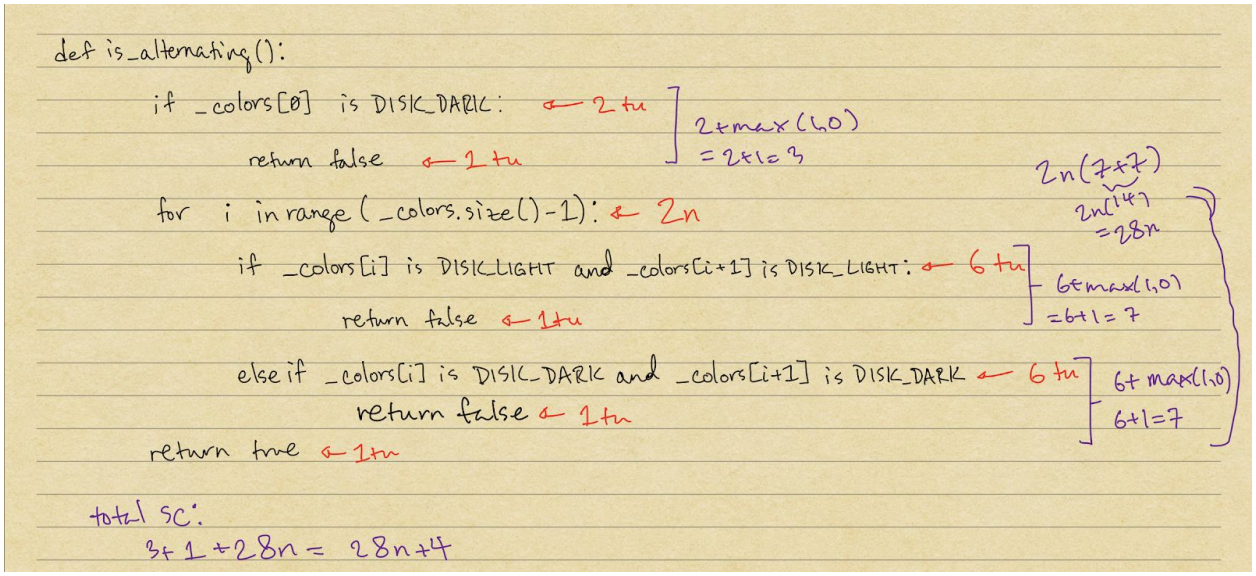
```
# sort_left_to_right algorithm pseudocode

def sort_left_to_right(disk_state before):
   assert(before.is_alternating())

   disk_state temp = before
   swap_count = 0

   for i in range(temp.light_count()):
       for j in range(temp.total_count - 1):
           if temp.get(j) is DISK_DARK and temp.get(j+1) is DISK_LIGHT:
               temp.swap(j)
               swap_count += 1

   return sorted_disks(temp, swap_count)
```

2. **sort_lawnmower algorithm:**

```
# sort_lawnmower pseudocode

def sort_lawnmower(disk_state before):
   assert(before.is_alternating())
   disk_state after = before
   swap_count = 0

   for i in range(after.light_count()/2):
       for j in range(after.total_count()-1):
           if after.get(j) is DISK_DARK and after.get(j+1) is DISK_LIGHT:
               after.swap(j)
               swap_count +=1

       for k in range(after.total_count()-2,0,-1):
           if after.get(k) is DISK_DARK and after.get(k+1) is DISK_LIGHT:
               after.swap(k)
               swap_count += 1

   return sorted_disks(after, swap_count)
```

Brief Proof Argument for time complexity:

Proof of Step Count for is_alternating():



Left to Right Algorithm Time Complexity Proof:

```
def sort_left_to_right(disk_state before):
    assert(before.is_alternating()) ← 28n+4
    disk_state temp = before ← 1 tu
    swap_count = 0 ← 1 tu
    for i in range(temp.light_count()): ← n-1
        for j in range(temp.total_count - 1): ← 2n-1
            if temp.get(j) is DISK_DARK and temp.get(j+1) is DISK_LIGHT: ← 6 tu
                temp.swap(j) ← 1 tu
                swap_count += 1 ← 1 tu
    return sorted_disks(temp, swap_count) ← 1 tu
```

28n + 4 + 2 = 28n+6

n-1 (16n-8)
= 16n² - 8n - 16n + 8
= 16n² - 24n + 8

2n-1 (8) = 16n-8

6 + max(2,0)
= 8

total sc:

28n+6 + 16n² - 24n + 8 + 1

= 16n² + 4n + 15

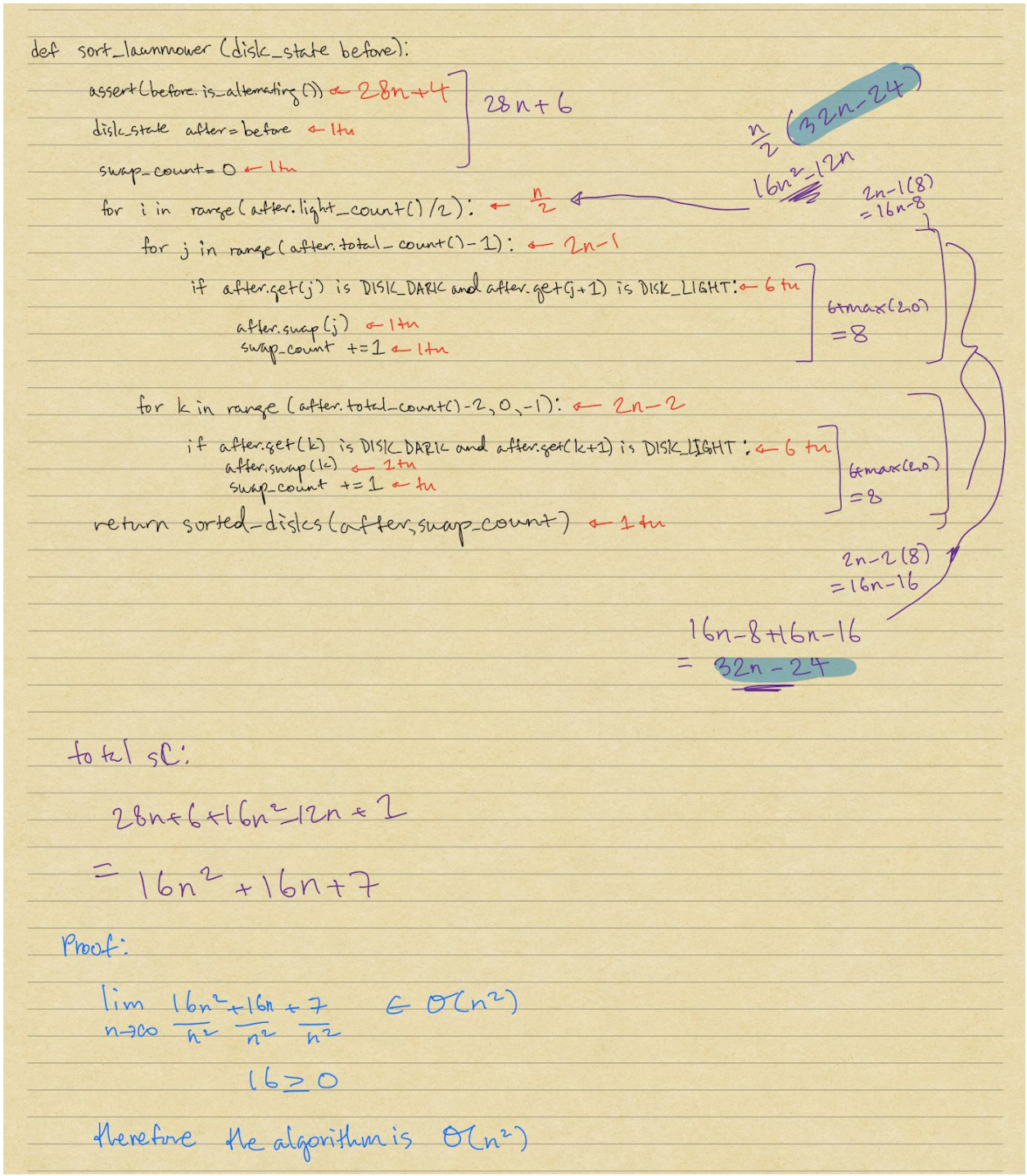
Proof:

$$\lim_{n \to \infty} \frac{16n^2}{n^2} + \frac{4n}{n^2} + \frac{15}{n^2} \quad \in O(n^2)$$

$16 \geq 0$

therefore the algorithm is $O(n^2)$

Lawnmower Algorithm Time Complexity Proof:



total sc:

$$28n+6+16n^2-12n+2$$

$$= 16n^2+16n+7$$

Proof:

$$\lim_{n\to\infty} \frac{16n^2}{n^2} + \frac{16n}{n^2} + \frac{7}{n^2} \quad \in O(n^2)$$

$$16 \geq 0$$

Therefore the algorithm is $O(n^2)$