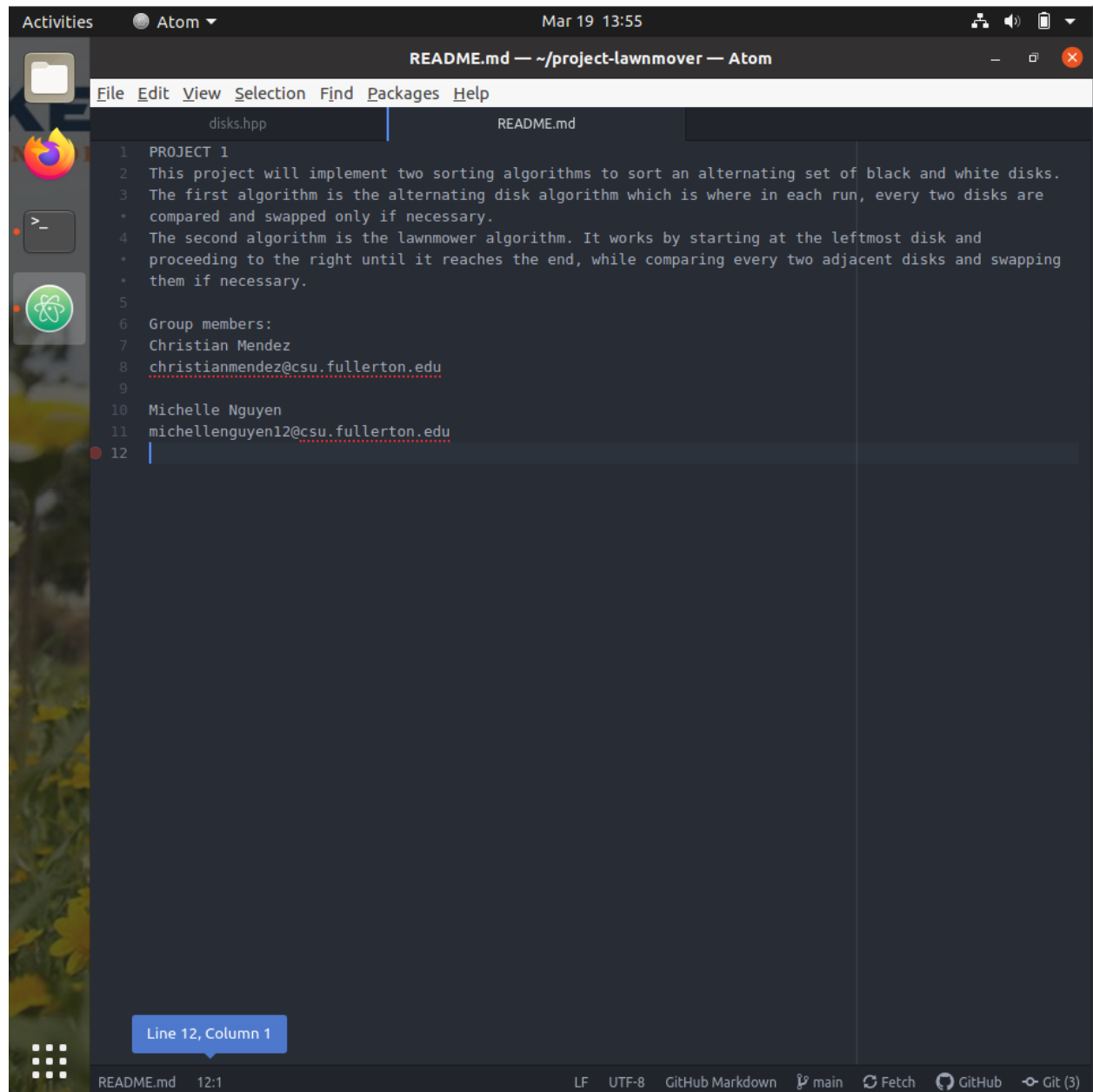


## Project 1

Christian Mendez, email: [christianmendez@csu.fullerton.edu](mailto:christianmendez@csu.fullerton.edu)

Michelle Nguyen, email: [michellenguyen12@csu.fullerton.edu](mailto:michellenguyen12@csu.fullerton.edu)



```
1 PROJECT 1
2 This project will implement two sorting algorithms to sort an alternating set of black and white disks.
3 The first algorithm is the alternating disk algorithm which is where in each run, every two disks are
4 * compared and swapped only if necessary.
5 The second algorithm is the lawnmower algorithm. It works by starting at the leftmost disk and
6 * proceeding to the right until it reaches the end, while comparing every two adjacent disks and swapping
7 * them if necessary.
8
9 Group members:
10 Christian Mendez
11 christianmendez@csu.fullerton.edu
12
13 Michelle Nguyen
14 michellenguyen12@csu.fullerton.edu
15
```

Line 12, Column 1

```

149 // Algorithm that sorts disks using the alternate algorithm.
150 sorted_disks sort_alternate(const disk_state& before) {
151     int numOfSwap = 0;
152     disk_state state = before;
153     for (size_t i = 0; i < state.total_count()-1; i++) {
154         for (size_t j = 0; j < state.total_count()-1; j++) {
155             if (state.get(j) % 2 == 1) {
156                 if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
157                     state.swap(j);
158                     numOfSwap++;
159                 }
160             }
161             if (state.get(j) % 2 == 0) {
162                 if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
163                     state.swap(j);
164                     numOfSwap++;
165                 }
166             }
167         }
168     }
169     return sorted_disks(disk_state(state), numOfSwap);
170 }
171
172 // Algorithm that sorts disks using the lawnmower algorithm.
173 sorted_disks sort_lawnmower(const disk_state& before) {
174     int numOfSwap = 0;
175     disk_state state = before;
176     for (size_t i = 0; i < state.total_count()-1; i++) {
177         for (size_t j = 0; j < state.total_count()-1; j++) {
178             if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
179                 state.swap(j);
180                 numOfSwap++;
181             }
182         }
183         for (size_t k = state.total_count()-1; k > 0; k--) {
184             if (state.get(k) == DISK_LIGHT && state.get(k-1) == DISK_DARK) {
185                 state.swap(k);
186                 numOfSwap++;
187             }
188         }
189     }
190     return sorted_disks(disk_state(state), numOfSwap);
191 }

```

Terminal Output:

```

student@tufflix-vm: ~/project-lawnmover
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
./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

```

## Step Count for Pseudocode sort\_alternate

numOfSwap = 0	// 1 tu
state = before	// 1 tu
for i = 0 to n - 1 do	// (n-1-0)+1 = n times
for j = 0 to n - 1 do	// (n-1-0)+1 = n times
if (j % 2 == 1)	// 2 tu
if (j == dark && j+1 == light)	// 4 tu
swap(j)	// 0 tu
numOfSwap++	// 1 tu
end if	
end if	
if (j % 2 == 0)	// 2 tu
if (j == dark && j+1 == light)	// 4 tu
swap(j)	// 0 tu
numOfSwap++	// 1 tu
end if	
end if	
end for	

SC<sub>a</sub> = 4 + max(1,0) = 5 tu

SC = 2 + max(0,0) = 2 tu

SC<sub>b</sub> = 4 + max(1,0) = 5 tu

SC = 2 + max(0,0) = 2 tu

SC = n \* n \* 14 + 1 + 1 = 14n<sup>2</sup> + 2 tu

### Proof for Pseudocode sort\_alterate

Show that  $14n^2+2$  belongs to  $O(n^2)$

$$F(n) = 14n^2 + 2$$

$$G(n) = n^2$$

Using def.  $f(n) \leq c \cdot g(n)$ ,  $n \geq n_0$

By def,  $14n^2 + 2 \leq c \cdot n^2$ ,  $n > n_0$

Let  $c = 40$  and  $n_0 = 1$

$$14n^2 + 2 \leq 40n^2, n > 1$$

$$14(1)^2 + 2 \leq 40(1)^2$$

$$16 \leq 40$$

This is true, hence  $14n^2 + 2$  belongs to the  $O(n^2)$  time complexity

### Step Count for Pseudocode sort\_lawnmower

```
numOfSwap = 0           // 1 tu
state = before           // 1 tu
for i = 0 to n - 1 do    // (n-1-0)+1 = n times
  for j = 0 to n - 1 do  // (n-1-0)+1 = n times
    if (j == dark && j+1 == light) // 4 tu
      swap(j)           // 0 tu
      numOfSwap++       // 1 tu
    end if
  end for
  for k = n-1 to 0 do    // (0-(n-1))+1 = n times
    if (k == light && k-1 == dark) // 4 tu
      swap(k)           // 0 tu
      numOfSwap++       // 1 tu
    end if
  end for
end for
```

$$SC = 4 + \max(1, 0) = 5 \text{ tu}$$

$$SC = 4 + \max(1, 0) = 5 \text{ tu}$$

$$SC = (n * n + n) * (5 + 5) + 1 + 1$$

$$SC = 10n^2 + 10n + 2$$

### Proof for Pseudocode sort\_lawnmower

Show that  $10n^2+10n+2$  belongs to  $O(n^2)$

$$F(n) = 10n^2 + 10n + 2$$

$$G(n) = n^2$$

Using def.  $f(n) \leq c \cdot g(n)$ ,  $n \geq n_0$

By def,  $10n^2 + 10n + 2 \leq c \cdot n^2$ ,  $n > n_0$

Let  $c = 40$  and  $n_0 = 1$

$10n^2 + 10n + 2 \leq 40n^2, n > 1$

$10(1)^2 + 10(1) + 2 \leq 40(1)^2$

$22 \leq 40$

This is true, hence  $10n^2 + 10n + 2$  belongs to the  $O(n^2)$  time complexity