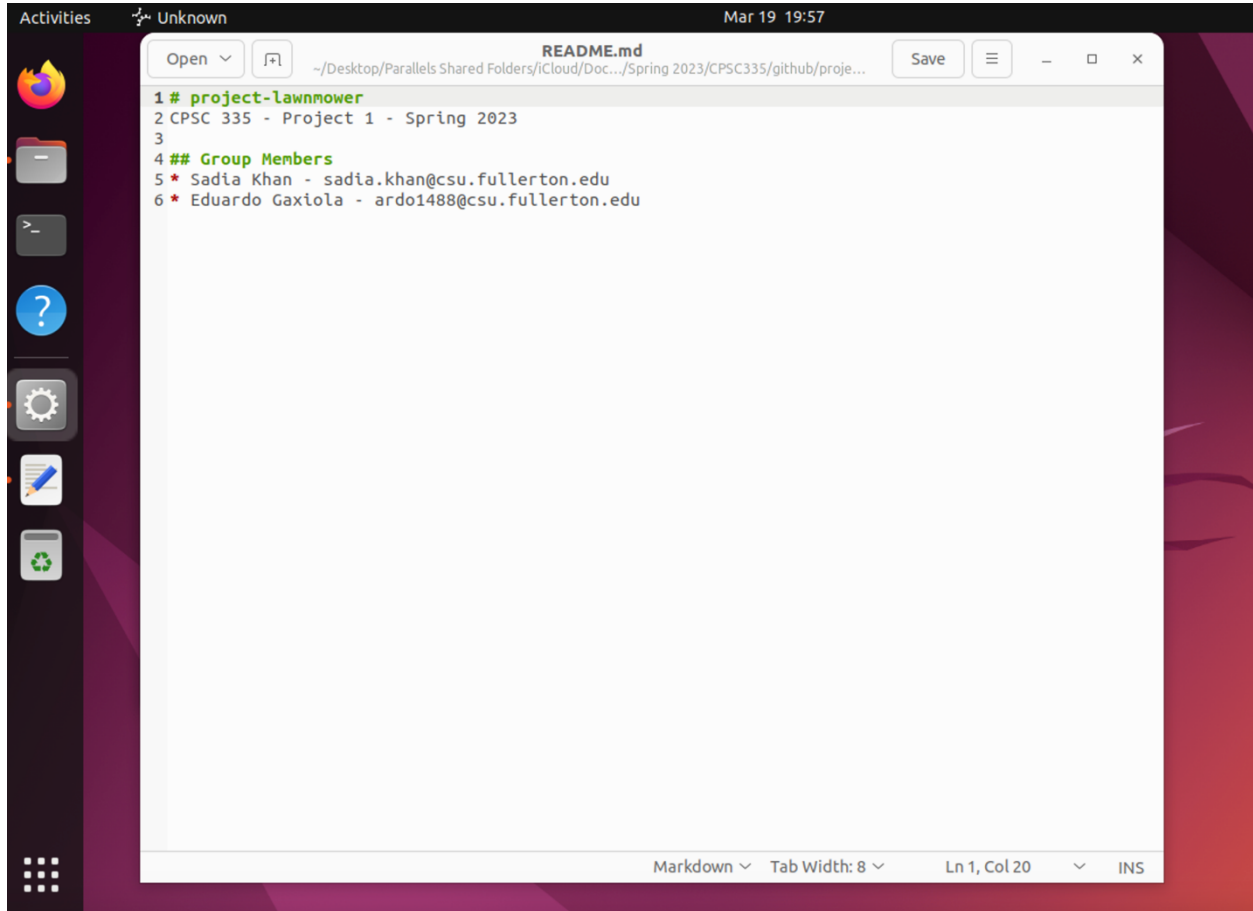


## Project Lawnmower (Project 1 for CPSC 335 course at CSUF – Spring 2023)

Group Members: Sadia Khan ([sadia.khan@csu.fullerton.edu](mailto:sadia.khan@csu.fullerton.edu)) and Eduardo Gaxiola ([ardo1488@csu.fullerton.edu](mailto:ardo1488@csu.fullerton.edu))

### Readme screenshot



## Screenshot of Code Compiling and Executing

```

1 # project-lawnmover
2 CPSC 335 - Project 1 - Spring 2023
3
4 ## Group Members
5 * Sadia Khan - sadia.khan@csu.fullerton.edu
6 * Eduardo Gaxiola - ardo1488@csu.fullerton.edu

parallels@ubuntu-linux-22-04-desktop: ~/Desktop/P...
parallels@ubuntu-linux-22-04-desktop: ~/Desktop/Parallels Shared Folders/Cloud/Documents/CPSC335/Spring 2023/github/project-lawnmover$ make
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

// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {
    int numOfSwap = 0;
    //record # of
    step swap
    disk_state state = before;
    size_t n = state.total_count();
    for(size_t i = 0; i < n; i++){
        // Loop through and sort
        for(size_t j = i%2; j < n - 1; j += 2){
            if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
                state.swap(j);
                numOfSwap++;
            }
        }
    }
    return sorted_disks(disk_state(state), numOfSwap);
}

// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state& before) {
    int numOfSwap = 0;
    disk_state state = before;
    size_t n = state.total_count();
    for(size_t i = 0; i < n/2 - 1; i++){
        if(i % 2 == 0){
            for(size_t j = 0; j < n-1; j+=1){
                if(state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT){
                    state.swap(j);
                    numOfSwap++;
                }
            }
        }
        else{
            for(size_t k = (n-1); k > 0; k-=1){
                if (state.get(k-1) == DISK_DARK && state.get(k) == DISK_LIGHT)
            }
        }
    }
}

```

Pseudocode listings, step count, and proof for time complexity:

**Alternate Algorithm:**

PseudoCode for Alternate Algorithm with step counts – Time Complexity:  $O(n^2)$ :

Input: alternating disks of size  $2n$

numOfSwap = 0 // 1 tu

for (i = 0 to  $2n$ , i++) do //  $2n-0+1 = 2n+1$  times

    for (j =  $i\%2$  to  $2n-1$ , j+=2) do //  $((2n-1)-0)/2 + 1 = n + 1/2$  or  $((2n-1)-1)/2 + 1 = n$

        if(disks[j] == dark and disks[j+1] == light) // 4 tu

            swap(disks[j], disks[j+1]) // 3 tu

            numOfSwap++ // 1 tu

return state, numOfSwap

*/\* Step Count (SC) for the Alternate Algorithm \*/*

s.c. for if inside inner for =  $4 + 4 = 8$

s.c. for inner for =  $\max((n + 1/2)*8, 8n) = 8n + 4$

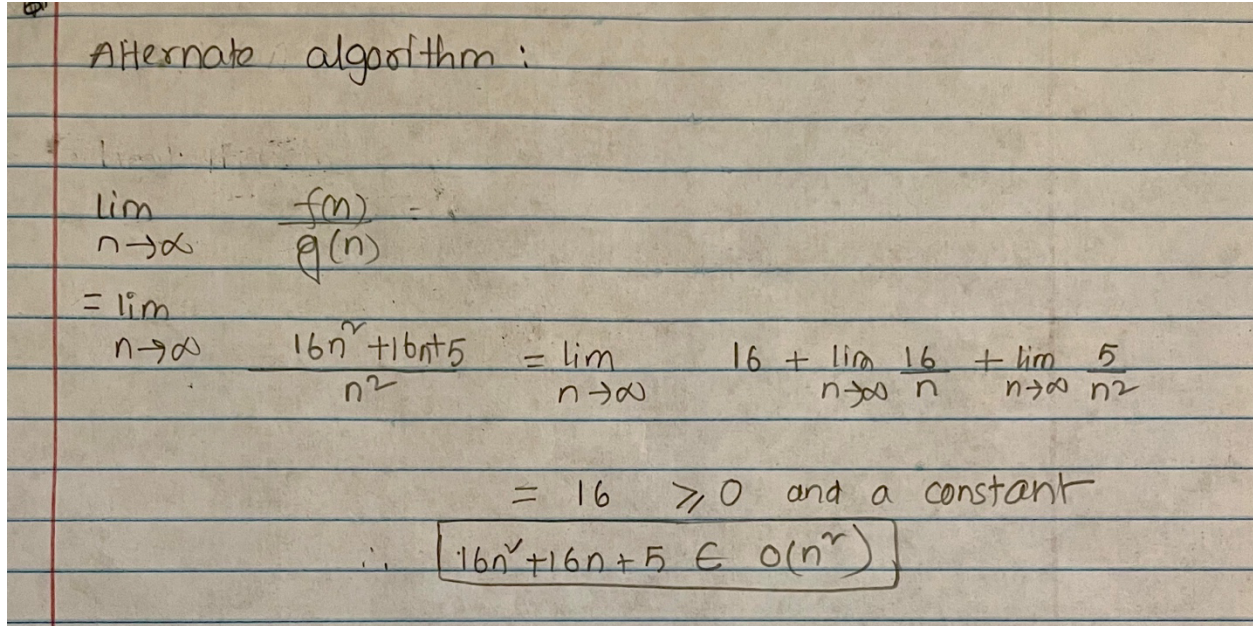
s.c. for outer for =  $(2n + 1) * (8n + 4) = 16n^2 + 16n + 4$

Total for Alternate Algorithm:

s.c. =  $1 + 16n^2 + 16n + 4 = 16n^2 + 16n + 5$

==> This leads to  $O(n^2)$

Proof:



Alternate algorithm:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$$
$$= \lim_{n \rightarrow \infty} \frac{16n^2 + 16n + 5}{n^2} = \lim_{n \rightarrow \infty} 16 + \lim_{n \rightarrow \infty} \frac{16}{n} + \lim_{n \rightarrow \infty} \frac{5}{n^2}$$
$$= 16 \geq 0 \text{ and a constant}$$

$\therefore 16n^2 + 16n + 5 \in O(n^2)$

**Lawnmower Algorithm:**

PseudoCode for Lawnmower Algorithm with step counts – Time Complexity:  $O(n^2)$ :

Input: alternating disks of size  $2n$

numOfSwap = 0 // 1 tu

for( i = 0 to n - 1) do //  $n-1-0+1 = n$  times

if( i % 2 == 0) then do // 2 tu

for( j = 0 to  $2n-2$ ) do //  $2n-2-0+1 = 2n-1$  times

if(disks[j] == dark and disks[j+1] == light) then do // 3 tu

swap(disks[j], disks[j+1]) // 3 tu

numOfSwap++ // 1 tu

else

for( k =  $2n-1$  to 1) do //  $2n-1+1 = 2n$

if(disks[k-1] == dark and disks[k] == light) then do // 3 tu

```
swap(disks[k-1], disks[k])    // 3 tu
numOfSwap++ // 1 tu
```

return state, numOfSwap

### Step Count for Lawnmower:

s.c inside the outer for loop:  $2 + \max(\max(6n-3, 14n-7), \max(6n, 14n)) = (14n + 2)$

sc. Outer loop :  $n(14n+2) = 14n^2 + 2n // O(n^2)$

$$\begin{aligned} & \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \\ &= \lim_{n \rightarrow \infty} \frac{14n^2 + 2n}{n^2} = \lim_{n \rightarrow \infty} 14 + \lim_{n \rightarrow \infty} \frac{2}{n} \\ &= 14 + 0 = 14 \geq 0 \text{ and a constant} \\ & \boxed{14n^2 + 2n \in O(n^2)} \end{aligned}$$