

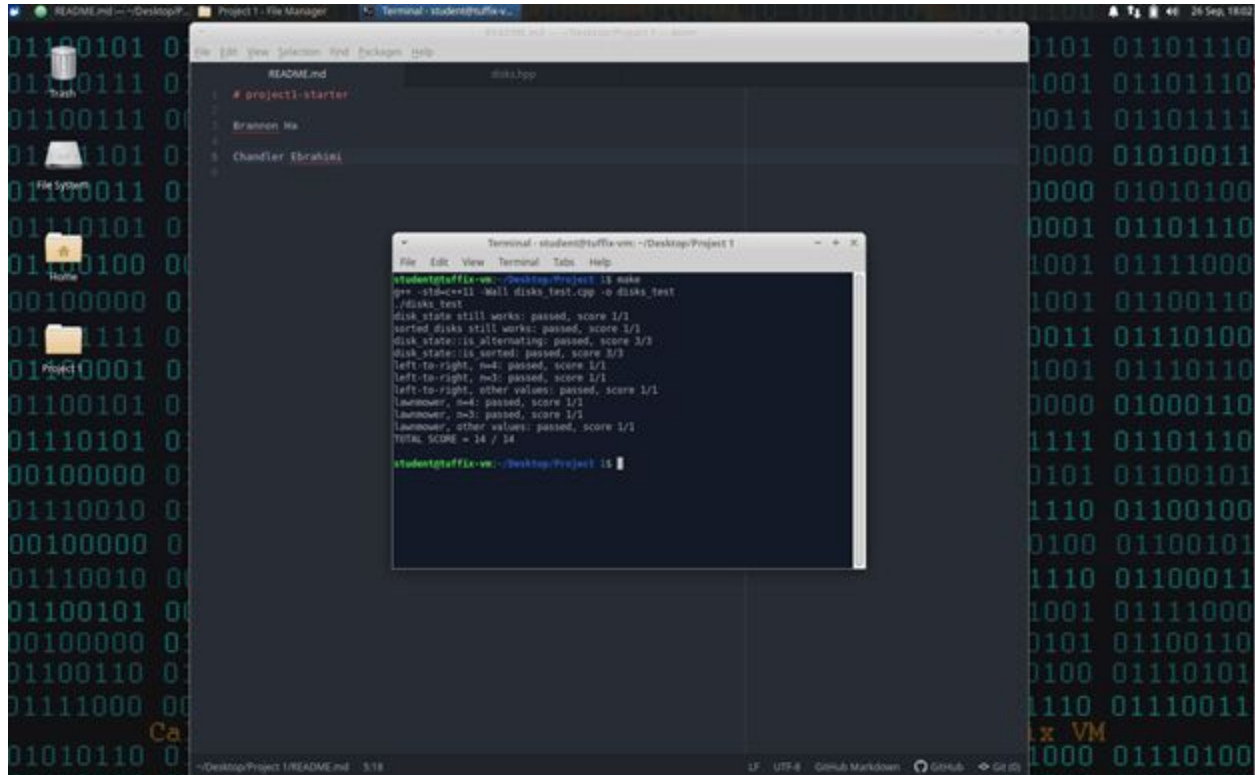
Project 1

Chandler Ebrahimi

csebra@csu.fullerton.edu

Brannon Ha

brannonha@csu.fullerton.edu



The screenshot shows a Linux desktop environment with a dark theme. In the background, a file manager window displays a directory structure with files like 'README.md', 'project1-starter', 'Brannon Ha', and 'Chandler Ebrahimi'. A terminal window is open in the foreground, showing the output of a 'make' command. The terminal output indicates that various tests passed, including 'disk state still works', 'sorted disks still works', 'disk state::is_alternating', 'disk state::is_sorted', 'left-to-right, m=1', 'left-to-right, m=3', 'left-to-right, other values', 'lawnmower, m=4', and 'lawnmower, m=3'. The total score is 14 / 14.

```
student@staff1x-vm: ~/Desktop/Project 1$ 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_alternating: passed, score 3/3
disk state::is_sorted: passed, score 3/3
left-to-right, m=1: passed, score 1/1
left-to-right, m=3: passed, score 1/1
left-to-right, other values: passed, score 1/1
lawnmower, m=4: passed, score 1/1
lawnmower, m=3: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 14 / 14
student@staff1x-vm: ~/Desktop/Project 1$
```

Left-to-Right Algorithm Pseudocode:

```
def sort_left_to_right(L)
```

```
    Set end equal to Total Disks
```

```
    Swaps = 0
```

```
    Set local to L
```

```
    For each i between 0 and end
```

```
        For each j between i and end
```

```
            If the current j index is a light disk and j+1 index is a dark disk
```

```
                Swap
```

```
                swaps++
```

```
    Return local
```

Lawnmower Algorithm Pseudocode:

def lawnmower(L)

Set end equal to Total Disks

Swaps = 0

Sorted = false

Set local to L

While sorted = false

 For each disk in L

 Compare local[i] to local[i+1]

 If out of order swap

 Swaps++

 For Max_disk_count to Zero

 Compare local[i] to local[i+1]

 If out of order swap

 Swaps++

 Check if sorted

 If sorted set sorted = true

Return local

Time Complexity of Algorithms:

The Left to Right algorithm has a its total step count to be equal to this equation:
 $1+1+1+(n*n*3) + 1 = 3n^2 + 3$ which can be simplified down to n^2 . Therefore the Left to Right algorithm falls under $O(n^2)$ complexity.

The Lawnmower algorithm has a total step count of:
 $1+1+1+(n-1)+1+1+(n-1)+1+1+1+1=n+7$ which can be simplified to n . Therefore the Lawnmower algorithm falls under $O(n)$ complexity.