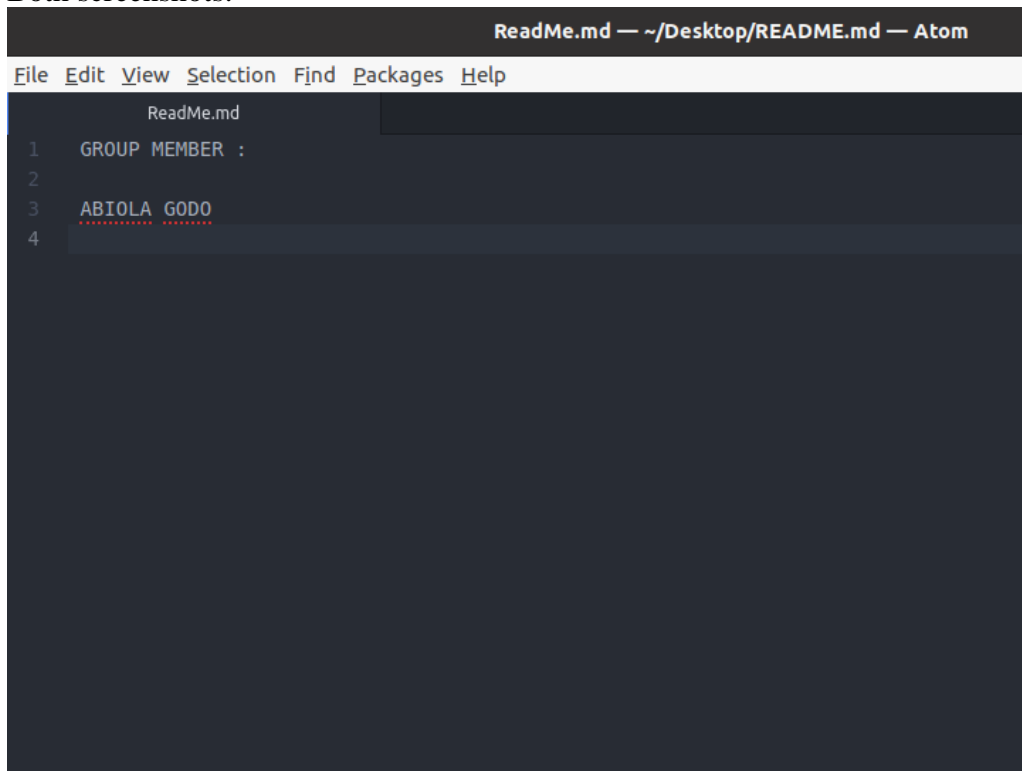


PROJECT 1: Implementing Algorithms

Submitted by:

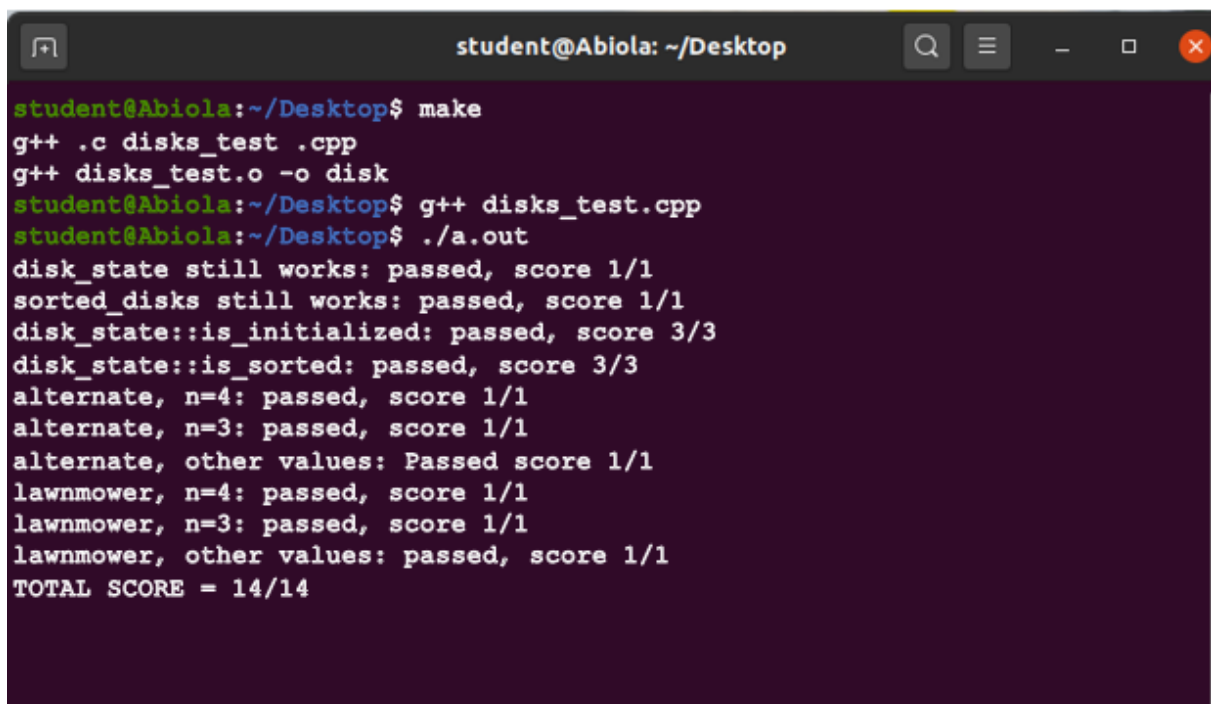
ABIOLA GODO
Abiola@csu.fullerton.edu

Both screenshots:



The screenshot shows the Atom text editor interface. The title bar at the top reads "ReadMe.md — ~/Desktop/README.md — Atom". Below the title bar is a menu bar with the following items: File, Edit, View, Selection, Find, Packages, and Help. The main editing area shows a file named "ReadMe.md" with the following content:

```
1 GROUP MEMBER :  
2  
3 ABIOLA GODO  
4 .....
```



The screenshot shows a terminal window with the title bar "student@Abiola: ~/Desktop". The terminal displays the following commands and output:

```
student@Abiola:~/Desktop$ make  
g++ .c disks_test .cpp  
g++ disks_test.o -o disk  
student@Abiola:~/Desktop$ g++ disks_test.cpp  
student@Abiola:~/Desktop$ ./a.out  
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
```

Following is a pseudocode of the Lawnmower Algorithm for the given problem.

Algorithm1: Lawnmower Algorithm

Input: $n \leftarrow$ a positive integer n

$L \leftarrow$ a list of $2n$ disks of alternating colors dark-light, starting with dark

Output: $L' \leftarrow$ a list of $2n$ disks, the first n disks are dark, the next n disks are light

$m \leftarrow$ an integer representing the number of swaps required

```
1.   $m \leftarrow 0$ 
2.   $L' \leftarrow L$ 
3.  For  $i = 1, 2, \dots, n/2$ :
4.      For  $j = 1, 2, \dots, 2n - 1$ :
5.          If  $L'[j] == \text{white}$  and  $L'[j+1] == \text{black}$ 
6.               $\text{swap}(L'[j], L'[j+1])$ 
7.               $m := m + 1$ 
8.          End If
9.      End For
10.     For  $j = 2n, 2n - 1, \dots, 2$ :
11.         If  $L'[j-1] == \text{white}$  and  $L'[j] == \text{black}$ 
12.              $\text{swap}(L'[j-1], L'[j])$ 
13.              $m := m + 1$ 
14.         End If
15.     End For
16. End For
17. Return  $L', m$ 
```

Complexity Analysis of the Lawnmower Algorithm:

Here the first loop (line 3) takes $n / 2$ iterations. Inside the first loop, there are two more loops (line 4, line 10). Both of them take $2n - 1$ iterations. The contents inside those loops have complexity of $O(1)$.

So the total complexity becomes:

$$\begin{aligned} & (n / 2) * ((2n - 1) + (2n - 1)) \\ \Rightarrow & (n / 2) * (4n - 2) \\ \Rightarrow & n * (2n - 1) \\ \Rightarrow & 2n^2 - n \end{aligned}$$

So we can say that the complexity of the Lawnmower Algorithm is $O(n^2)$.

Following is a pseudocode of the Alternate Algorithm for the given problem.

Algorithm2: Alternate Algorithm

Input: $n \leftarrow$ a positive integer n

$L \leftarrow$ a list of $2n$ disks of alternating colors dark-light, starting with dark

Output: $L' \leftarrow$ a list of $2n$ disks, the first n disks are dark, the next n disks are light

$m \leftarrow$ an integer representing the number of swaps required

```

1.   $m \leftarrow 0$ 
2.   $L' \leftarrow L$ 
3.  For  $i = 1, 2, \dots, n$ :
4.      If  $i \% 2 == 1$ 
5.          For  $j = 1, 3, \dots, 2n - 1$ :
6.              If  $L'[j] == \text{white}$  and  $L'[j+1] == \text{black}$ 
7.                   $\text{swap}(L'[j], L'[j+1])$ 
8.                   $m := m + 1$ 
9.              End If
10.         End For
11.     End If
12.     Else
13.         For  $j = 2, 4, \dots, 2n - 2$ :
14.             If  $L'[j] == \text{white}$  and  $L'[j+1] == \text{black}$ 
15.                  $\text{swap}(L'[j], L'[j+1])$ 
16.                  $m := m + 1$ 
17.             End If
18.         End For
19.     End Else
20. End For
21. Return  $L', m$ 

```

Complexity Analysis of the Alternate Algorithm:

Here the first loop (line 3) takes n iterations. Inside the first loop, there are two more loops (line 5, line 13). Both of them take n and $n - 1$ iterations correspondingly. The contents inside those loops have complexity of $O(1)$.

So the total complexity becomes:

$$\begin{aligned}
 & n * (n + (n - 1)) \\
 \Rightarrow & n * (2n - 1) \\
 \Rightarrow & 2n^2 - n
 \end{aligned}$$

So we can say that the complexity of the Alternate Algorithm is $O(n^2)$.