CPSC 335 - Section 07
Project 1: Lawnmower Problem

Bryan Garcia, PieIsManly@csu.fullerton.edu
Bryan Rivas, brivas100@csu.fullerton.edu

Pseudocode

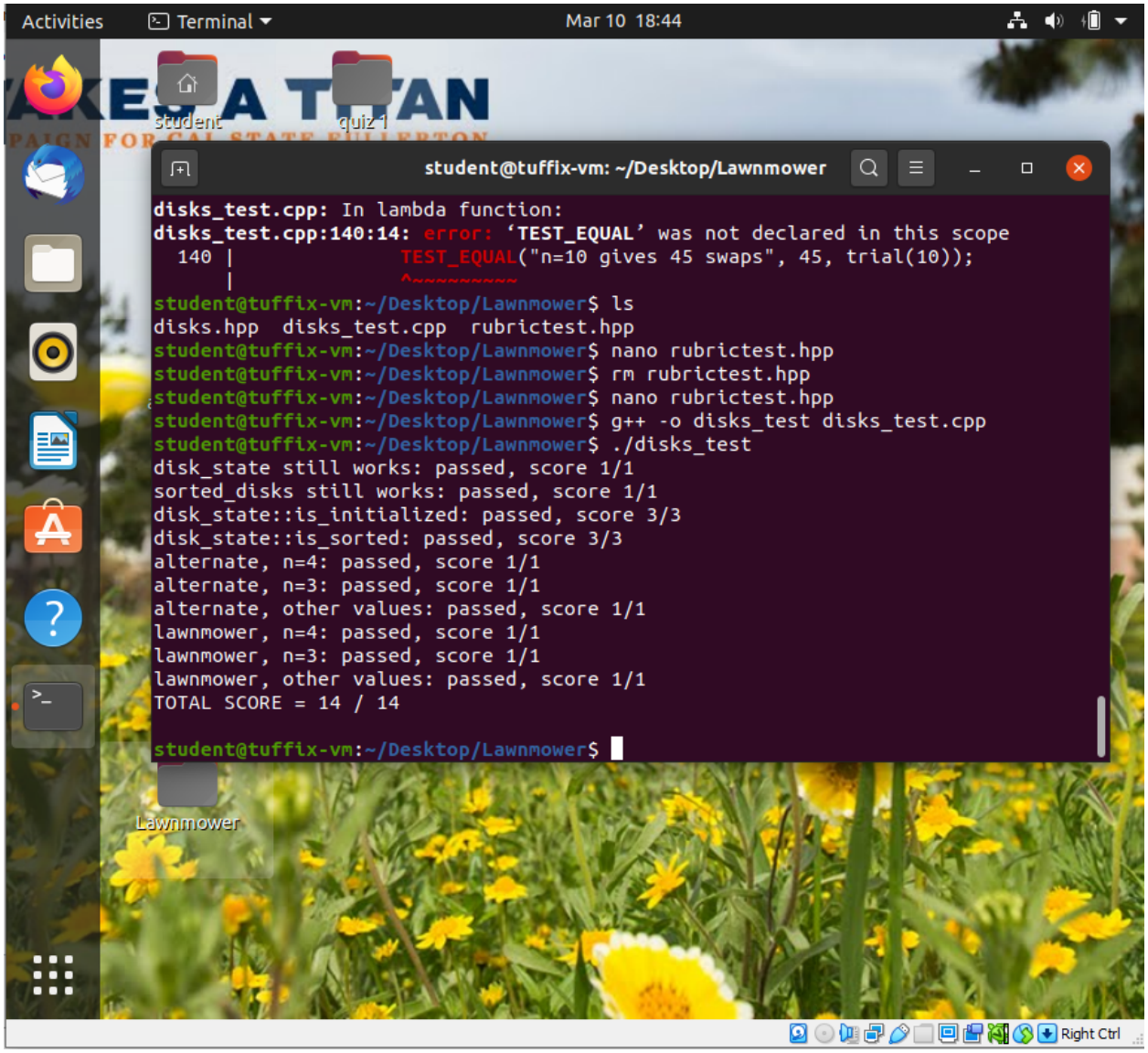
```
Def Sort_Alternate():
    While Disks is not sorted:
        For i = 0 from i to n, 2 step:                      -> ((n - 0 + 1)/2) * 4 = 2(n+1) Tu
            If (Disks[i] == Dark && Disks[i+1] == Light)    -> 3 + (Max(1,0)) = 4 Tu
                Swap Disks[i] and Disks[i+1]                -> 1 Tu
        For i = 1 from i to n-1, 2 step:                    -> ((n -1 - 1 + 1)/2) * 4 = 2(n+1) Tu
            If (Disks[i] == Dark && Disks[i+1] == Light)    -> 3 Tu + (Max(1,0)) = 4 Tu
                Swap Disks[i] and Disks[i+1]                -> 1 Tu
Total Step Count: 4(n+1) Tu

Def Lawnmower():
    While Disks is not sorted:
        For i = 0 from i to n-1:                            -> ((n-1 -0 +1)/1) * 4 = 4n Tu
            If (Disks[i] == Dark && Disks[i+1] == Light)    -> 3 + (Max(1,0)) = 4Tu
                Swap Disks[i] and Disks[i+1]                -> 1 Tu
        For i = n from i to 1, -1 step:                     -> ((1 - n +1)/-1) * 4 = 4(n-2) Tu
            If (Disks[i] == Light && Disks[i-1] == Dark)    -> 3 Tu + (Max(1,0))
                Swap Disks[i] and Disks[i-1]                -> 1 Tu
Total Step Count: 4n - 8 + 4n = 8n - 8 = 8(n-1) Tu
```

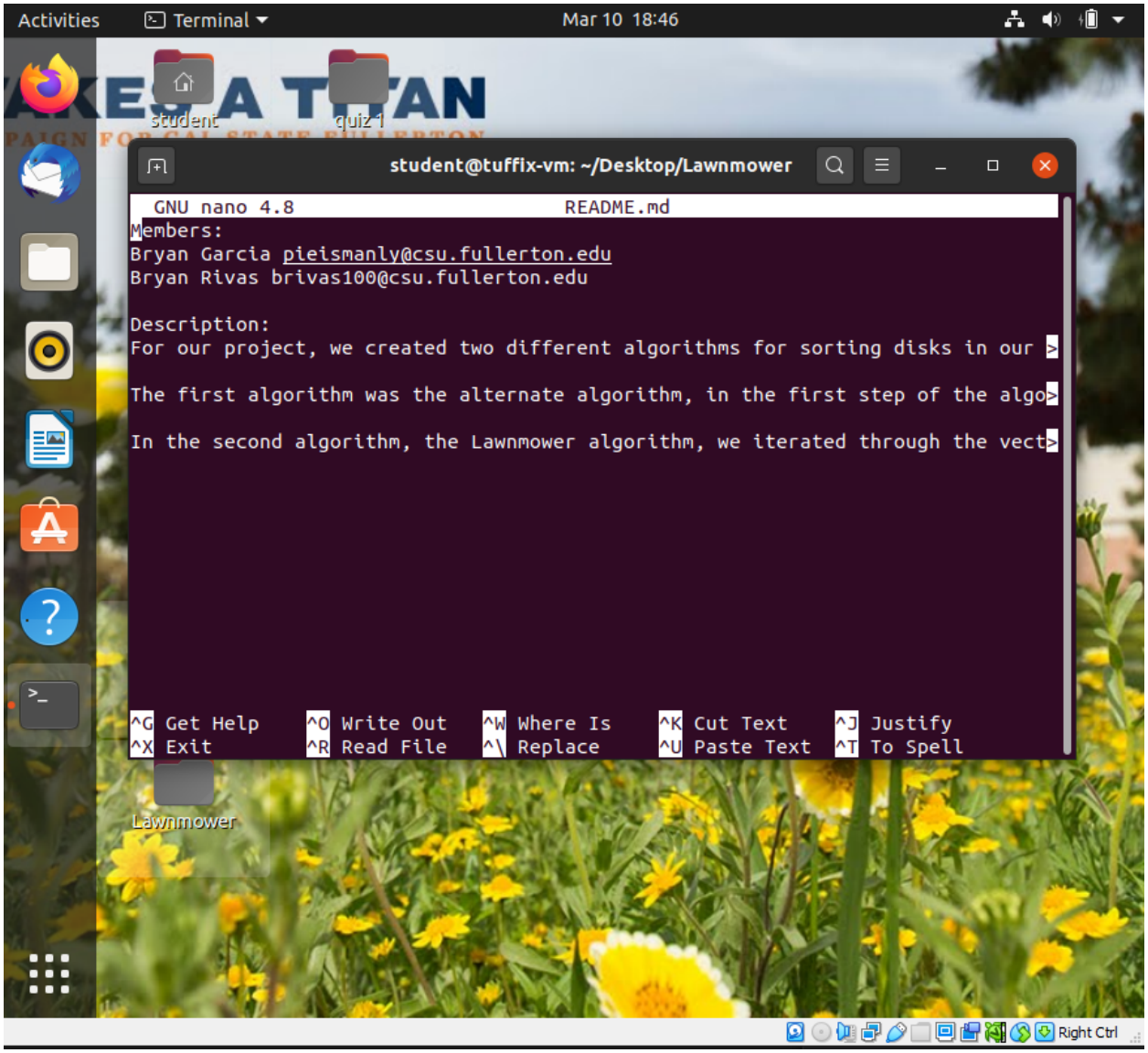
Time Complexity Explanation:

For our Sort_Alternate() Algorithm, We ended up with a step count of 4(n+1) Time Units, with n being our only variable, it would be a Time-Complexity of O(N).

For our Lawnmower Algorithm, We ended up with a step count of 8(n-1) Time Units, Having n again as our only variable, it would also be a Time-Complexity of O(N).

Screenshots


Activities
Terminal
Mar 10 18:44
student
quiz 1
student@tuffix-vm: ~/Desktop/Lawnmower
disks_test.cpp: In lambda function:
disks_test.cpp:140:14: error: 'TEST_EQUAL' was not declared in this scope
140 | TEST_EQUAL("n=10 gives 45 swaps", 45, trial(10));
| ^~~~~~~~~~
student@tuffix-vm:~/Desktop/Lawnmower$ ls
disks.hpp disks_test.cpp rubrictest.hpp
student@tuffix-vm:~/Desktop/Lawnmower$ nano rubrictest.hpp
student@tuffix-vm:~/Desktop/Lawnmower$ rm rubrictest.hpp
student@tuffix-vm:~/Desktop/Lawnmower$ nano rubrictest.hpp
student@tuffix-vm:~/Desktop/Lawnmower$ g++ -o disks_test disks_test.cpp
student@tuffix-vm:~/Desktop/Lawnmower$ ./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
student@tuffix-vm:~/Desktop/Lawnmower$
Lawnmower
Right Ctrl


Activities
Terminal
Mar 10 18:46
student
quiz 1
student@tuffix-vm: ~/Desktop/Lawnmower
GNU nano 4.8 README.md
Members:
Bryan Garcia pieismanly@csu.fullerton.edu
Bryan Rivas brivas100@csu.fullerton.edu
Description:
For our project, we created two different algorithms for sorting disks in our >
The first algorithm was the alternate algorithm, in the first step of the algo>
In the second algorithm, the Lawnmower algorithm, we iterated through the vect>
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell
Lawnmower
Right Ctrl