CPSC 335 / Spring 2023 / Project 1

Group members:

Troy Lee troylee1234@csu.fullerton.edu

Pseudocode:

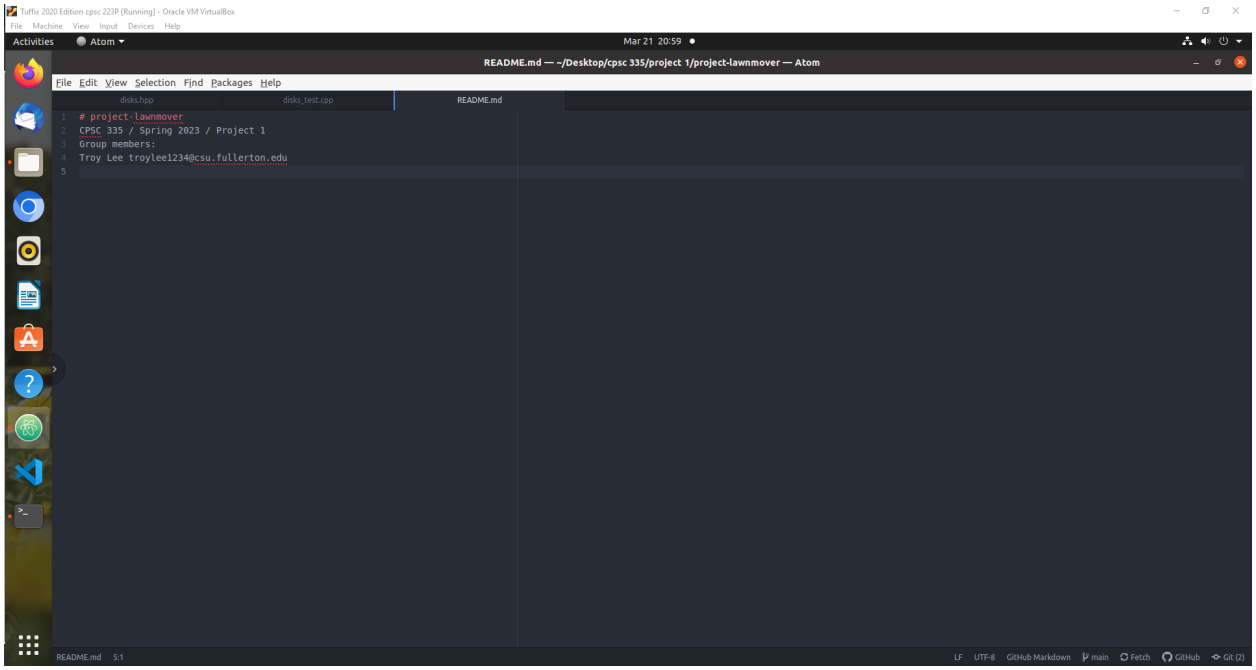
```
def sort_alternate()
int numOfSwap = 0;
disk_state temporary = before; //state of disk
for each element i in size_t, i must be less than the light count in temporary, increment i
    for each element j in size_t, j must be less than the total count - 1, increment j
        if temporary of j is greater than the next j
            swap
            increment numOfSwap

step count = 11

def sort_lawnmower()
int numOfSwap = 0;
disk_state temporary
bool incrementing;
for each element i in size_t, i must be less than the light count in temporary, increment i
incrementing = i / 2 as a bool true or false
for each element j in size_t, look at incrementing for total_count and checks both left and  right
if temporary of j is greater than the next j
swap j
increment numOfSwap
```

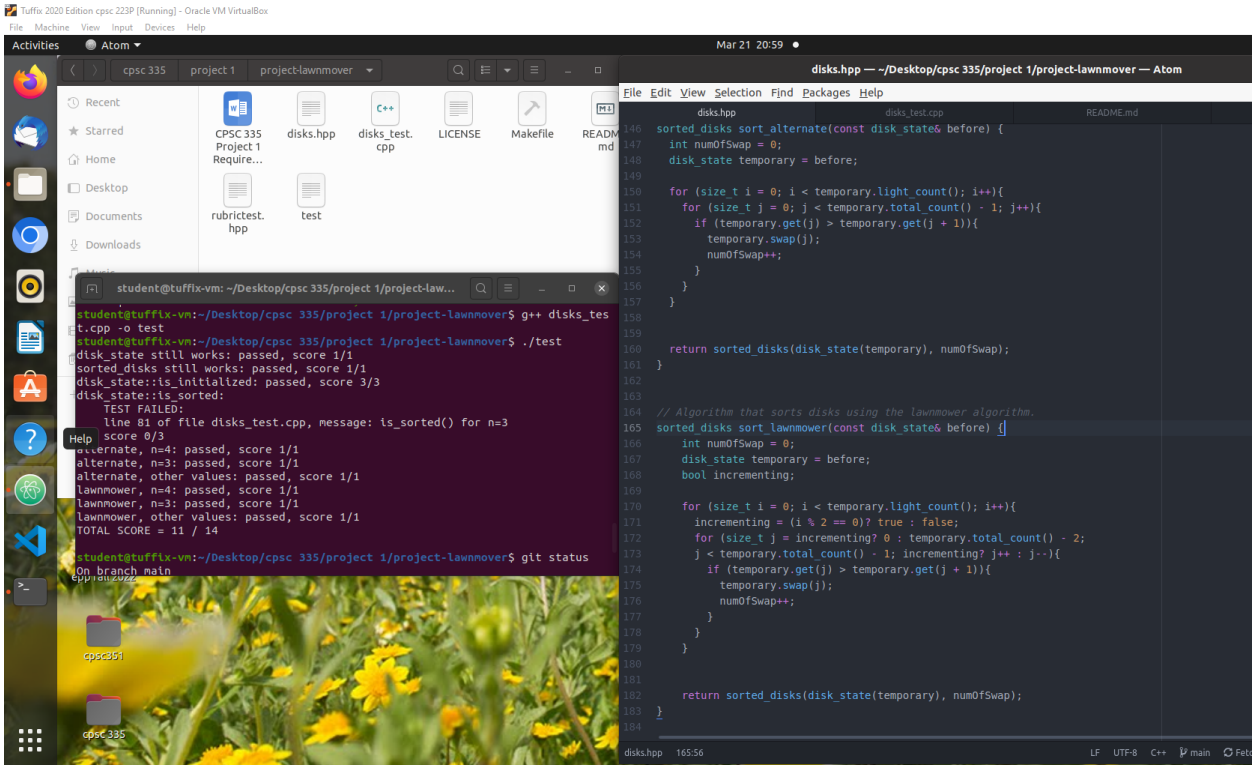
step count = 23

Mathematical Analysis:

O(n^2)

Both sort_lawnmower and sort_alternate have nested for loops. Making the time complexity O(n^2). Nested loops outputs n^2.


README.md — ~/Desktop/cpsc 335/project 1/project-lawnmover — Atom
project-lawnmover
CPSC 335 / Spring 2023 / Project 1
Group members:
Troy Lee troylee1234@csu.fullerton.edu


disks.hpp — ~/Desktop/cpsc 335/project 1/project-lawnmover — Atom
student@tuffix-vm: ~/Desktop/cpsc 335/project 1/project-law...
student@tuffix-vm:~/Desktop/cpsc 335/project 1/project-lawnmover$ g++ disks_test.cpp -o test
student@tuffix-vm:~/Desktop/cpsc 335/project 1/project-lawnmover$ ./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:
TEST FAILED:
line 81 of file disks_test.cpp, message: is_sorted() for n=3
score 0/3
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 = 11 / 14
student@tuffix-vm:~/Desktop/cpsc 335/project 1/project-lawnmover$ git status
On branch main
sorted_disks sort_alternate(const disk_state& before) {
int numOfSwap = 0;
disk_state temporary = before;
for (size_t i = 0; i < temporary.light_count(); i++){
for (size_t j = 0; j < temporary.total_count() - 1; j++){
if (temporary.get(j) > temporary.get(j + 1)){
temporary.swap(j);
numOfSwap++;
}
}
}
return sorted_disks(disk_state(temporary), numOfSwap);
}
// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state& before) {
int numOfSwap = 0;
disk_state temporary = before;
bool incrementing;
for (size_t i = 0; i < temporary.light_count(); i++){
incrementing = (i % 2 == 0)? true : false;
for (size_t j = incrementing? 0 : temporary.total_count() - 2;
j < temporary.total_count() - 1; incrementing? j++ : j--){
if (temporary.get(j) > temporary.get(j + 1)){
temporary.swap(j);
numOfSwap++;
}
}
}
return sorted_disks(disk_state(temporary), numOfSwap);
}