Project 1: Implementing Algorithms

Group members
Andrew Lau - andrewlau2019@csu.fullerton.edu
Tommy Le - tommyle@csu.fullerton.edu

Pseudocode for lawnmower

```
def sort_lawnmower() {
    int swap = 0

    for i = 0 to (n+1)/2 step 1 do
        for j = 0 to 2n - 1  step 1 do
            if(get(j) == DISK_DARK && get(j+1) == DISK_LIGHT)
                then swap(j)
                        swap++
            endif
        endfor
        for j = 2n - 1 to 0 step -1 do
            if(get(j) == DISK_LIGHT && get(j - 1) == DISK_DARK)
                then swap(j-1)
                        swap++
            endif
        endfor
    endfor

    return(swap)
}
```

Pseudocode for Alternate

```
def sort_alternate() {
    int swap = 0

    for i = 0 to n + 1 do
        for j = i % 2 to 2n - 1 step 2 do
            if(get(j) == DISK_DARK && get(j+1) == DISK_LIGHT)
                then swap(j)
                        swap++
            endif
        endfor
    endfor

    return(swap)
}
```

Step Count for lawnmower

```
def sort_lawnmower() {
    int swap = 0                                                    // 1 tu

    for i = 0 to (n+1)/2 step 1 do                  // [(n+1)/2 - 1 = (n-1)/2]
                                                    ((n-1)/2 - 0/1) + 1 = (n+1)/2 times

        for j = 0 to 2n - 1 step 1 do                   //[2n-1 - 1 = 2n-2]
                                                        ((2n-2 - 0/1) + 1 =  2n - 1 times)
            if(get(j) == DISK_DARK && get(j+1) == DISK_LIGHT)       // 14 tu
                then swap(j)                                        // 9 tu
                    swap++                                          // 1 tu
            endif
        endfor
        for j = 2n - 1 to 0 step -1 do                  //[0 - 1 = -1/-1 = 1]
                                                        ((1 - (2n-1)/-1) + 1 = 2n -1 times
            if(get(j) == DISK_LIGHT && get(j-1) == DISK_DARK)       // 14 tu
                then swap(j-1)                                      // 10 tu
                    swap++                                          // 1 tu
            endif
        endfor
    endfor

    return(swap)
}

void swap(size_t left_index) {
   assert(is_index(left_index));                    // 3 tu
   auto right_index = left_index + 1;               // 2 tu
   assert(is_index(right_index));                   // 3 tu
   std::swap(_colors[left_index], _colors[right_index]);        // 1 tu
 }
                                                                Total: 3 + 2 + 3 + 1 = 9 tu
_____________________________________________
bool is_index(size_t i) const {
   return (i < total_count()); // 2 tu
 }

size_t total_count() const {
```

```
   return _colors.size(); // `1 tu
  }

disk_color get(size_t index) const {
   assert(is_index(index));                    // 3 tu
   return _colors[index];
  }
```

SC = 1 * (n+1)/2 * [(2n -1) * (14 + 9 + 1)] * [(2n-1) * (14 + 10 + 1)]
= 1 * (n+1)/2 *([(2n-1) * 24] + [(2n-1) * (25)]) = 1 * (n+1)/2 * [48n-24 + 50n -25]
= (n+1)/2 * [98n-49] = 49n^2 +49n/2 - 49/2

Proof

$49n^2 + 49n/2 - 49/2 \in O(n^2)$

Lim n->infinity f(n)/g(n) = L
Lim n-> infinity (49n^2 +49n/2 - 49/2)/(n^2)
Lim n-> infinity (49n^2/n^2) + (49n/2/n^2) + (49/2/n^2)
Lim n-> infinity 49 + Lim n-> infinity 49n^3/2 + Lim n-> infinity 49n^2/2
49 + 0 + 0 = 49

$49 >= 0$ therefore $49n^2 + 49n/2 + 49/2 \in O(n^2)$

Step Count for Alternate

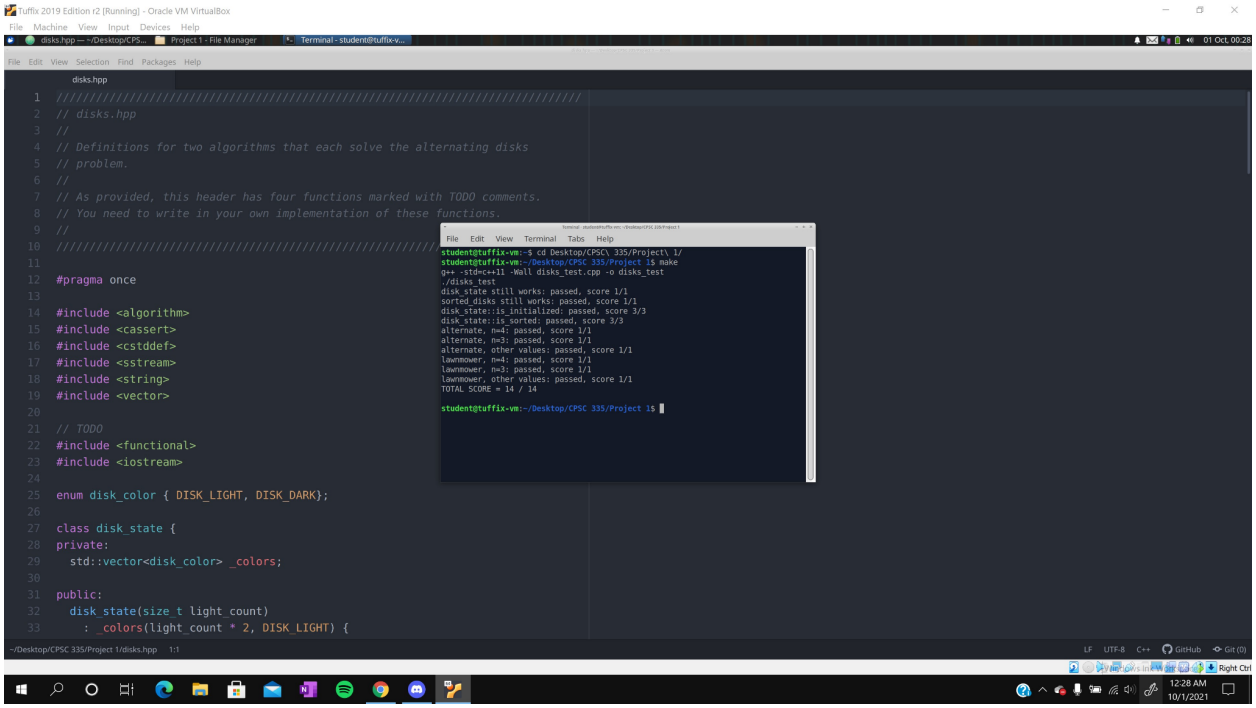
```
def sort_alternate() {
    int swap = 0                                                    // 1 tu

    for i = 0 to n + 1 step 1 do
        for j = i % 2 to 2n - 1 step 2 do
            if(get(j) == DISK_DARK && get(j+1) == DISK_LIGHT)       // 14 tu
                then swap(j)                                        // 9 tu
                     swap++                                         // 1 tu
            endif
        endfor
    endfor

    return(swap)
}
```

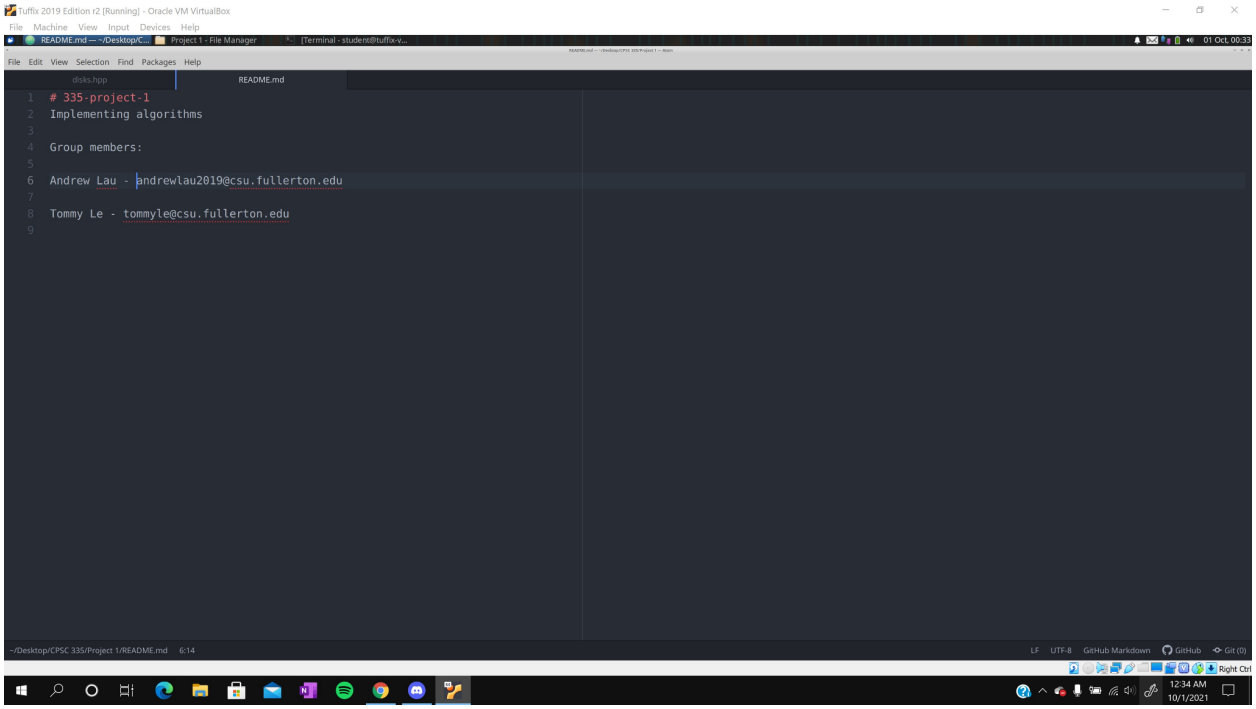
S.C = $\sum_{i=0}^{n} \sum_{j=0}^{2n-2} \{step\ 2\}\ 24 = 24 * [(2n - 2 - 0/2) + 1] * [(n - 0/1) + 1]$= 24 * (n) * (n+1) = 24n * (n+1) = 24n^2 +24n

Proof

24n^2 + 24n ∈ O(n^2)

Lim n->infinity f(n)/g(n) = L
Lim n->infinity (24n^2 + 24n)/n^2
Lim n->infinity 24 + Lim n-> infinity 24/n
24 + 0 = 24
Since it is a constant and it's greater than 0, therefore 24n^2 + 24n ∈ O(n^2)


Tuffix 2019 Edition r2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
disks.hpp — ~/Desktop/CPS...
Project 1 - File Manager
Terminal - student@tuffix-v...
01 Oct, 00:28
File Edit View Selection Find Packages Help
disks.hpp
//
// disks.hpp
//
// Definitions for two algorithms that each solve the alternating disks
// problem.
//
// As provided, this header has four functions marked with TODO comments.
// You need to write in your own implementation of these functions.
//
//
#pragma once
#include <algorithm>
#include <cassert>
#include <cstddef>
#include <sstream>
#include <string>
#include <vector>
// TODO
#include <functional>
#include <iostream>
enum disk_color { DISK_LIGHT, DISK_DARK};
class disk_state {
private:
std::vector<disk_color> _colors;
public:
disk_state(size_t light_count)
: _colors(light_count * 2, DISK_LIGHT) {
File Edit View Terminal Tabs Help
student@tuffix-vm:~$ cd Desktop/CPSC\ 335/Project\ 1/
student@tuffix-vm:~/Desktop/CPSC 335/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_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/CPSC 335/Project 1$
~/Desktop/CPSC 335/Project 1/disks.hpp 1:1
LF UTF-8 C++ GitHub Git (0)
Right Ctrl
12:28 AM
10/1/2021


Tuffix 2019 Edition r2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
README.md — ~/Desktop/C...
Project 1 - File Manager
[Terminal - student@tuffix-v...
01 Oct, 00:33
File Edit View Selection Find Packages Help
disks.hpp
README.md
335-project-1
Implementing algorithms
Group members:
Andrew Lau - andrewlau2019@csu.fullerton.edu
Tommy Le - tommyle@csu.fullerton.edu
~/Desktop/CPSC 335/Project 1/README.md 6:14
LF UTF-8 GitHub Markdown GitHub Git (0)
Right Ctrl
12:34 AM
10/1/2021