

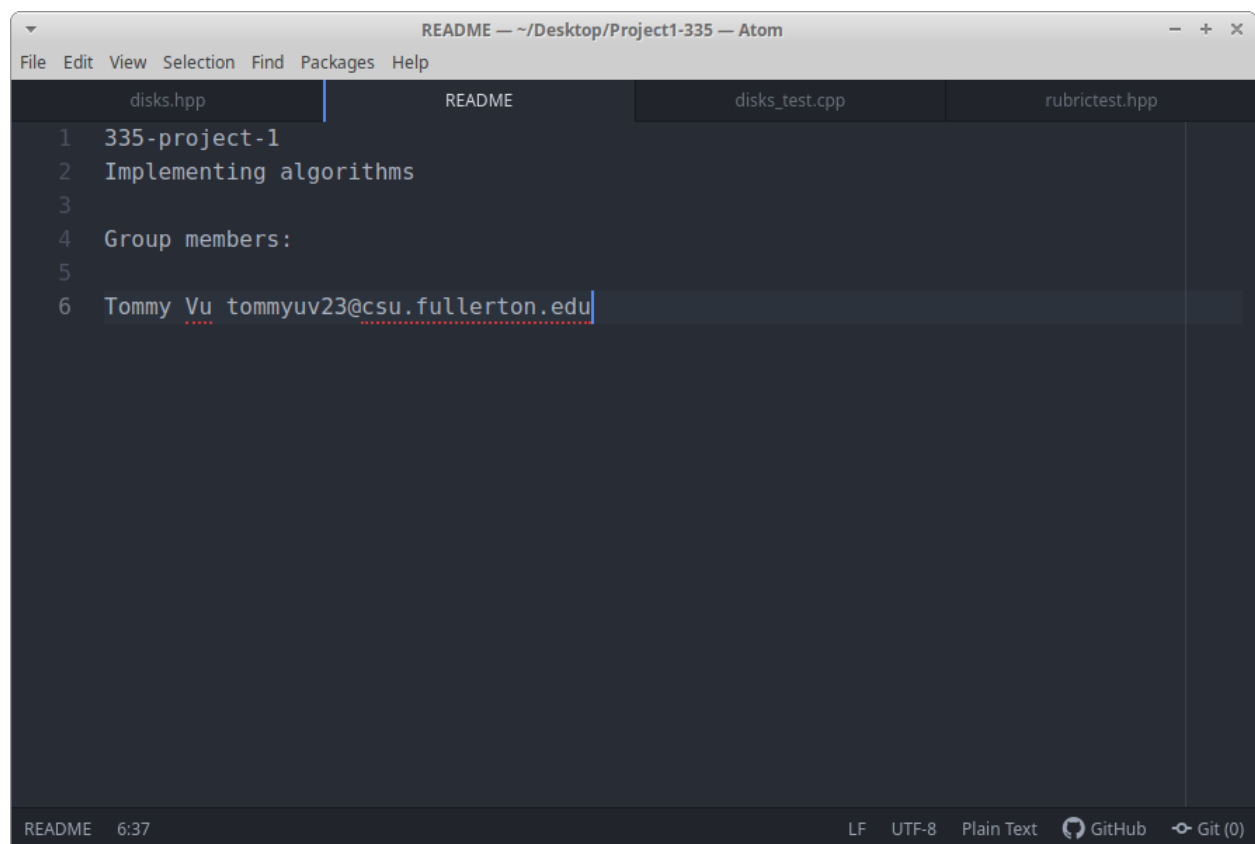
Tommy Vu

tommyuv23@csu.fullerton.edu

Project 1 - CPSC 335-01

Professor Bein

Screenshots:



The screenshot shows an Atom text editor window titled "README — ~/Desktop/Project1-335 — Atom". The window has a menu bar with "File", "Edit", "View", "Selection", "Find", "Packages", and "Help". Below the menu bar is a tab bar with four tabs: "disks.hpp", "README", "disks_test.cpp", and "rubrictest.hpp". The "README" tab is active, showing a dark-themed code editor with the following content:

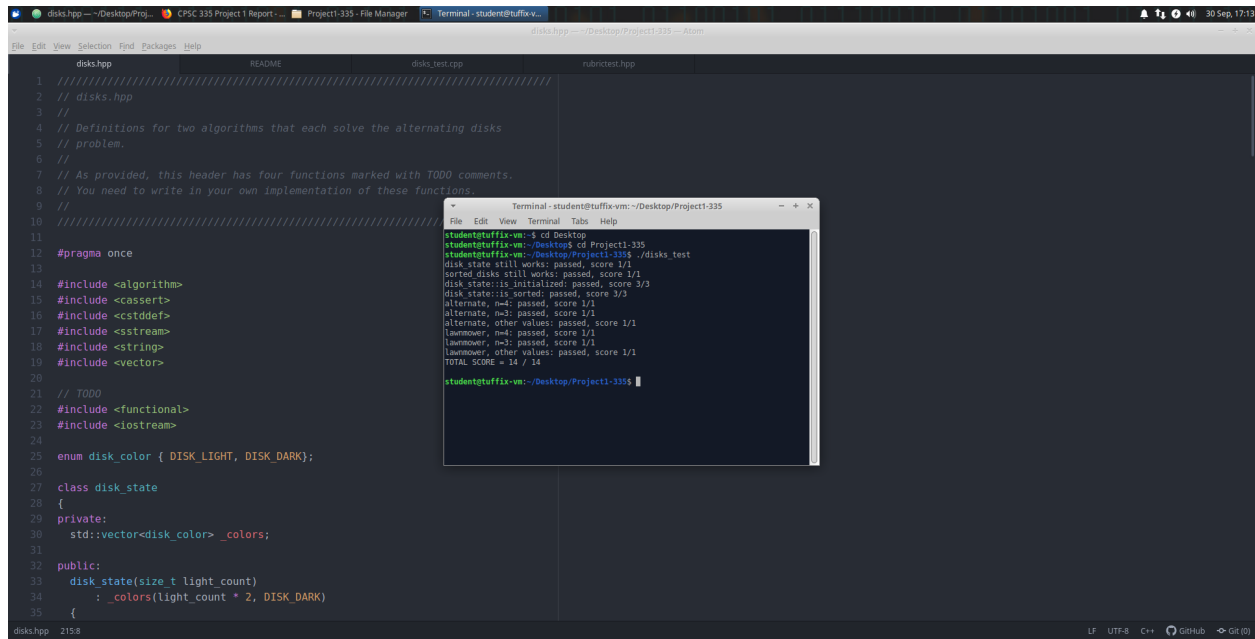
```
1 335-project-1
2 Implementing algorithms
3
4 Group members:
5
6 Tommy Vu tommyuv23@csu.fullerton.edu
```

The text "Tommy Vu tommyuv23@csu.fullerton.edu" on line 6 is highlighted with a blue selection bar. The status bar at the bottom of the window displays "README 6:37" on the left and "LF UTF-8 Plain Text GitHub Git (0)" on the right.

```
Terminal - student@tuffix-vm: ~/Desktop/Project1-335
File Edit View Terminal Tabs Help
student@tuffix-vm:~$ cd Desktop
student@tuffix-vm:~/Desktop$ cd Project1-335
student@tuffix-vm:~/Desktop/Project1-335$ ./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/Project1-335$
```

Full screen (It is a bit small, that's why I included the two separate ones)



The screenshot shows the Atom text editor with a project named 'Project1-335'. The main editor window displays the 'disks.hpp' header file, which contains comments and C++ code for a disk scheduling problem. The code includes standard headers like `<algorithm>`, `<cassert>`, `<csddef>`, `<sstream>`, `<string>`, and `<vector>`. It defines an enumeration `disk_color` with values `DISK_LIGHT` and `DISK_DARK`, and a class `disk_state` with a private member `std::vector<disk_color> _colors` and a public constructor `disk_state(size_t light_count)` that initializes `_colors` with `2 * DISK_DARK` elements.

A terminal window is open in the foreground, showing the output of running the program. The output displays the results of various tests, including 'sorted disks still works', 'disk state::is initialized', 'disk state::is sorted', 'alternate, n=4', 'alternate, other values', 'lawnmower, n=4', 'lawnmower, n=2', and 'lawnmower, other values'. Each test is marked as 'passed' with a score of 1/1. The total score is 14 / 14.

```
1 // disks.hpp
2 //
3 //
4 // Definitions for two algorithms that each solve the alternating disks
5 // problem.
6 //
7 // As provided, this header has four functions marked with TODO comments.
8 // You need to write in your own implementation of these functions.
9 //
10 //
11
12 #pragma once
13
14 #include <algorithm>
15 #include <cassert>
16 #include <csddef>
17 #include <sstream>
18 #include <string>
19 #include <vector>
20
21 // TODO
22 #include <functional>
23 #include <iostream>
24
25 enum disk_color { DISK_LIGHT, DISK_DARK};
26
27 class disk_state
28 {
29 private:
30     std::vector<disk_color> _colors;
31
32 public:
33     disk_state(size_t light_count)
34         : _colors(light_count * 2, DISK_DARK)
35     {
```

```
student@tuffix-vm:~/Desktop/Project1-335
student@tuffix-vm:~/Desktop/Project1-335 ./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 1/1
alternate, n=4: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, n=2: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 14 / 14
student@tuffix-vm:~/Desktop/Project1-335
```

Pseudo Code for Alternating Algorithm

```

int alg = 0 1
for i = 0 to n
    for n = i to 2n-1
        if (n > n+1) 2 + (max(3,2))    2 +(max(3,2))
            Swap                        3
            algo = algorithm + 1        2
        endif
    endfor
endfor

return sorted_disks(before, alg);

```

Step Count + Proof for Alternating Algorithm

$$\begin{aligned}
 & \sum_{i=0}^n \sum_{n=i}^{2n-1} 5 \\
 & \sum_{i=0}^n 5(2n-1) + 1 \\
 & \sum_{i=0}^n 10n - 4 \\
 & 10 \sum_{i=0}^n n - \sum_{i=0}^n 4 \\
 & 10 \cdot \frac{n(n+1)}{2} - 4n \\
 & \frac{10n^2 + 10n}{2} - 4n \\
 & 5n^2 + 5n - 4n
 \end{aligned}$$

$$= 5n^2 + 5n$$

Big O

Show that $f(n) = 5n^2 + n \in O(n^2)$

Find $C > 0$ and $n_0 \geq 0$ S.T. $5n^2 + n \leq (C)n^2 \quad \forall \quad n \geq n_0$

$$C = 5 + 1 = 6, n = 1$$

$$5n^2 + n \leq 6n^2$$

$$6n^2 - 5n^2 - n \geq 0$$

$$n^2 - n \geq 0$$

$$0 \geq 0$$

Yes, it belongs to $O(n^2)$.

Pseudo Code for Lawnmower Algorithm

Note: $n = \text{light_count}$
 $2n = \text{total count}$

```

int alg = 0;
for int i = 0 to n
    for j = i to 2n - 1
        if (j > (j+1))
            swap()
            algo = algo + 1
    Endfor
    for j = 2n - 1 to 0
        if (j < (j - 1))
            swap()
            algo = algo + 1
    endfor
endfor
return sorted_disks()

```

Step Count + Proof for Lawnmower Algorithm

$$\begin{aligned}
 \sum_{i=0}^n \sum_{t=i}^{2n-1} 5 &= \sum_{i=0}^n 5(2n-1) \\
 \sum_{i=0}^n 10(n) &- \sum_{i=0}^n 5 \\
 10 \cdot \frac{n(n+1)}{2} &- 5n \\
 \frac{10n^2 + 10n}{2} &- 5n \\
 5n^2 + 5n - 5n & \\
 &= 5n
 \end{aligned}$$

Big O

Show that $f(n) = 15n^2 + 20n + 10 \in O(n^2)$

Find $C > 0$ and $n_0 \geq 0$ S.T. $15n^2 + 20n + 10 \leq (C)n^2 \quad \forall \quad n \geq n_0$

$$C = 15 + 20 + 10 = 45, n_0 = 0$$

$$15n^2 + 20n + 10 \leq 45n^2$$

$$45n^2 + 15n^2 + 20n + 10 \geq 0$$

$$45n^2 + 15n^2 + 20n + 10$$

$$n \geq 0$$

Yes, it belongs to $O(n^2)$