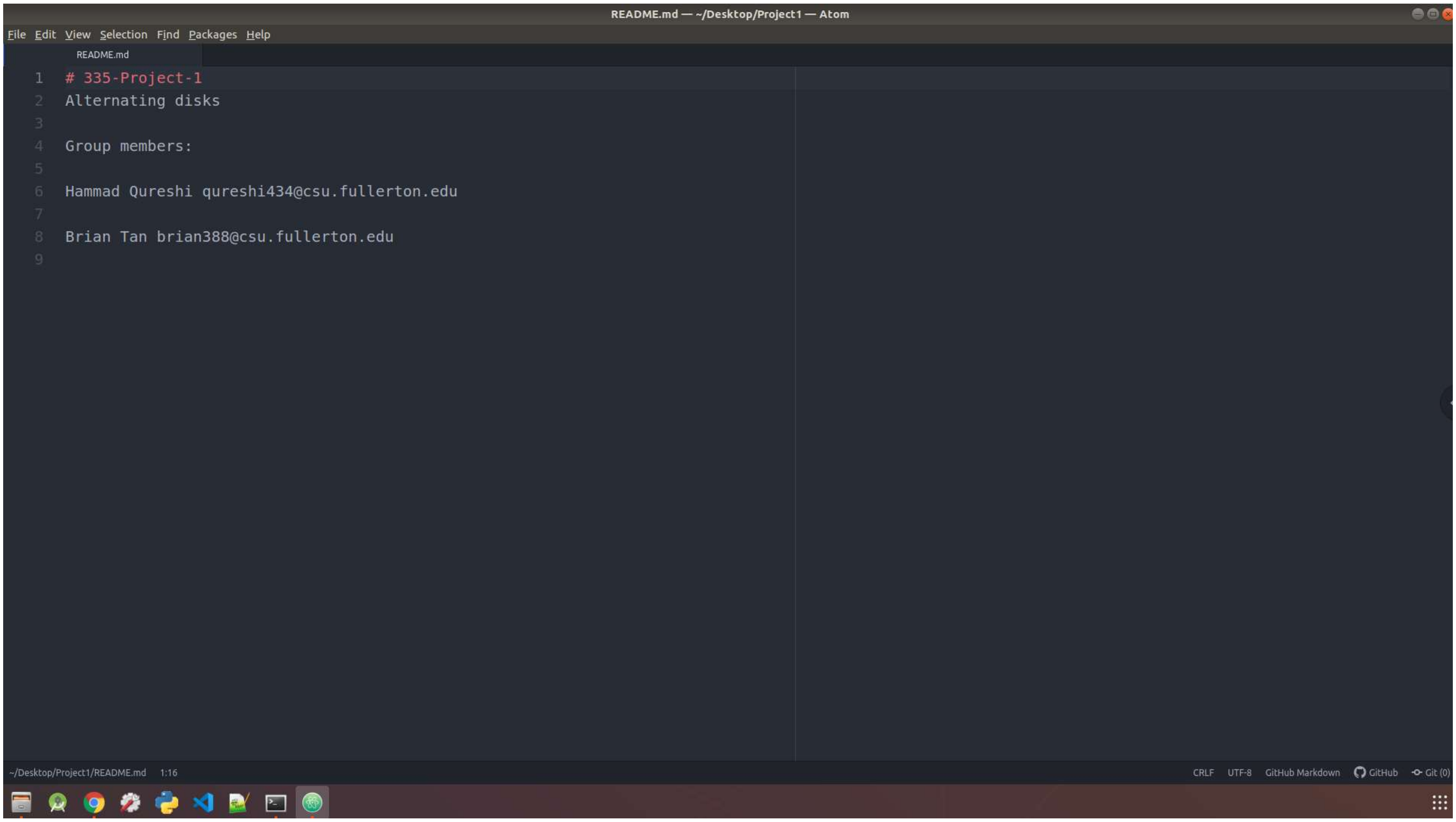
Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

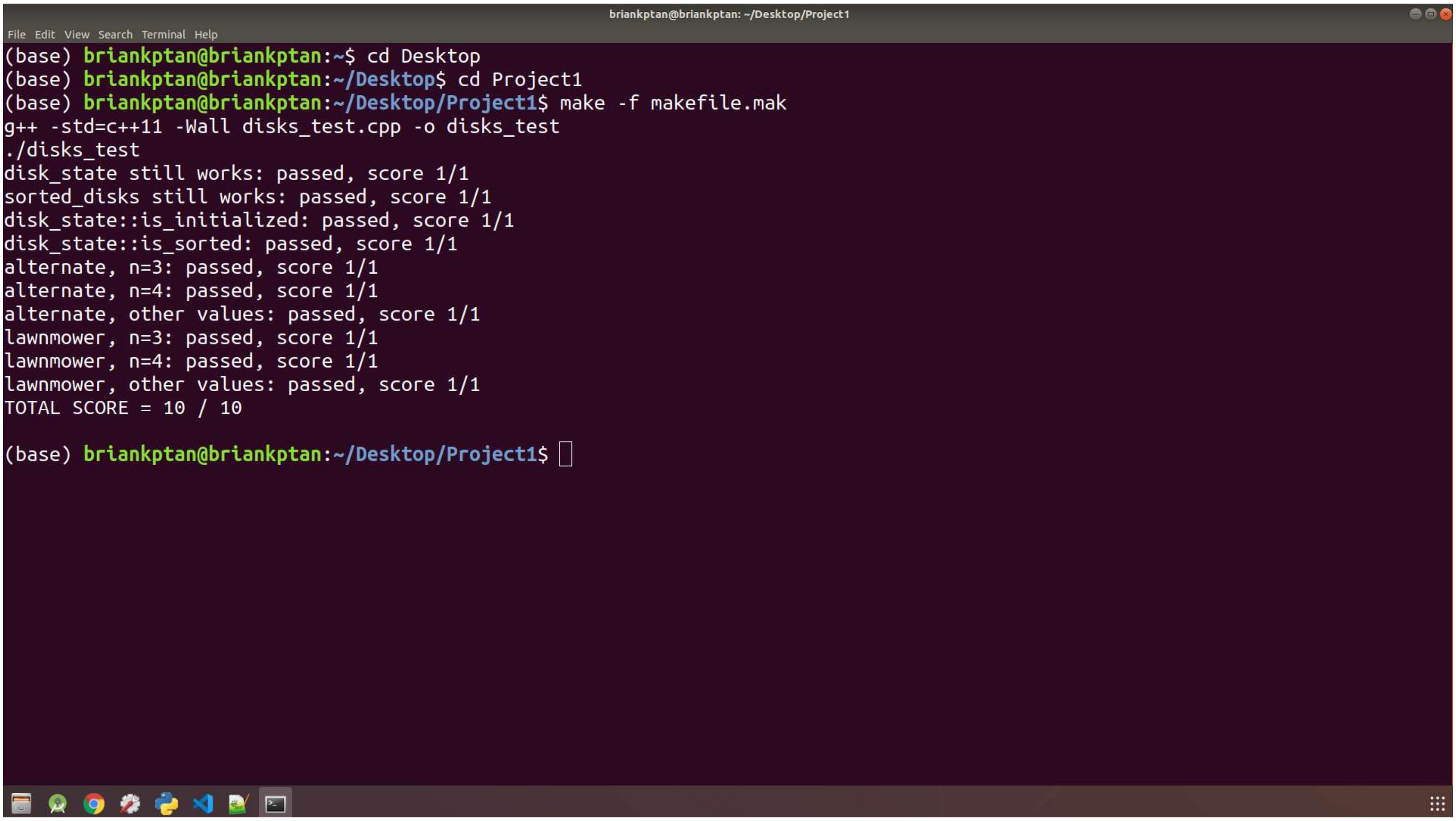

README.md — ~/Desktop/Project1 — Atom
File Edit View Selection Find Packages Help
README.md
~/Desktop/Project1/README.md 1:16
CRLF UTF-8 GitHub Markdown GitHub Git (0)

```
# 335-Project-1
Alternating disks

Group members:

Hammad Qureshi qureshi434@csu.fullerton.edu

Brian Tan brian388@csu.fullerton.edu
```


briankptan@briankptan: ~/Desktop/Project1
File Edit View Search Terminal Help

```
(base) briankptan@briankptan:~$ cd Desktop
(base) briankptan@briankptan:~/Desktop$ cd Project1
(base) briankptan@briankptan:~/Desktop/Project1$ make -f makefile.mak
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 1/1
disk_state::is_sorted: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, n=4: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 10 / 10

(base) briankptan@briankptan:~/Desktop/Project1$ 
```

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

Project 1

Pseudocode for Alternate Algorithm

```
sort_alternate(before)

after = before              // 1tu
swap_count = 0              //1tu
run_count = 0               //1tu

if after.issorted() == true then    // 8n + 16 tu
    return sequence unchanged
else
    if after.is_initialized() == true then      // $\frac{5n}{2} + 7$ tu
        for i = 0 to n-1 do                 //outer for loop
            if run_count % 2 != 0 then                      //2 tu
                for j = 0 to 2n-1, step 2, do               //inner loop 1
                    if color[j] != color[j+1] then      //2tu
                        swap()                          //5tu
                        swap_count++                    //1tu
                    endif
                endfor

            else
                for j = 1 to 2n-2, step 2, do               //inner loop 2
                    if color[j] != color[j+1] then      //2tu
                        swap()                          //5tu
                        swap_count++                    //1tu
                    endif
                endfor
            endif
            run_count++                                     //1tu
        endfor
    else
        return sequence unchanged
    endif
endif
```

// is_initialized() function step count calculated to be $\frac{5n}{2} + 7$ tu

// is_sorted() function step count calculated to be 8n + 16 tu

//swap() function determined to be 5tu

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

Proof and Time Complexity for Alternate Algorithm

(Independent for loops)

outer for loop:

using $\frac{end-start}{step} + 1$:

$\frac{n-1-0}{1} + 1 = n$

inner for loop 1:

$\frac{2n-1}{2} + 1 = n + \frac{1}{2}$

multiply with the if statement time units: $8(n + \frac{1}{2}) = 8n + 4$ //2tu + 5tu + 1tu = 8tu

inner for loop 2:

$\frac{2n-2-1}{2} + 1 = n - \frac{1}{2}$

multiply with the if statement time units: $8(n - \frac{1}{2}) = 8n - 4$

Combined:

outer for loop * (2 + max(inner loop 1, inner loop 2) + 1) //+1 because of the run_count increment

= n * (2 + max(8n + 4, 8n - 4) + 1)

= n * (2 + (8n + 4) + 1)

= n * (8n + 7)

= $8n^2 + 7n$

Add the time units outside the outer loop:

$8n^2 + 7n + 3 + \frac{5n}{2} + 7 + 8n + 16$

$= 8n^2 + \frac{35n}{2} + 26$

highest power is 2. Therefore, algorithm efficiency class is $O(n^2)$.

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

Proof by definition

Prove $8n^2 + \frac{35n}{2} + 26 \in \mathrm{O}(\mathrm{n}^2)$

choose C > 0 and a value n_0 >= 0 such that:

$8n^2 + \frac{35n}{2} + 26 <= C * n^2 \ \forall\ n >= n_0$

choose $C = 8 + \frac{35}{2} + 26 \cong 52$

$8n^2 + \frac{35n}{2} + 26 <= 52n^2$, true for n >= 1

choose $n_0 = 1$

$\therefore\ 8n^2 + \frac{35}{2} + 26 \in \mathrm{O}(\mathrm{n}^2)$

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

Pseudocode for Lawnmower Algorithm

```
sort_lawnmower(before)

after = before                  //1tu
swap_count = 0                  //1tu
temp_n = total_n - 1            //2tu

if after.issorted() == true then        //8n + 16 tu
    return sequence unchanged

else
    if after.is_initialized() == true then                  // $\frac{5n}{2} + 7\ tu$
        for i = 0 to n-1 do                                 //outer loop
            for j = i to 2n-1, step 2, do                   //inner loop 1
                if color[j] != color[j+1] then              //2tu
                    swap()                                  //5tu
                    swap_count++                            //1tu
                endif
            endfor

            temp_n--                                        //1tu

            if temp_n < i then                              //1tu
                for k = 2n-1 down to i, step 2, do          //inner loop 2
                    if color[k] != color[k-1] then          //2tu
                        swap()                              //5tu
                        swap_count++                        //1tu
                    endif
                endfor
            endif
        endfor
    else
        return sequence unchanged
    endif
endif
```

//swap() function determined to be 5tu

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

Proof and Time Complexity for Lawnmower Algorithm

Proof Time Complexity

(Dependent for loops)

outer for loop:
part of this loop has 1 decrement, and 1 if statement. So multiply by 2tu

$$\sum_{i=0}^{n-1} 2$$

inner loop 1:
1 if statement with 2tu operations, and carries out 5tu function and 1tu increment. So, 2+5+1= 8tu

since step of 2, use $\frac{end-star}{step}$ + 1.

8 ($\frac{2n-1-i}{2}$ + 1) = 8 (n + $\frac{1-i}{2}$)

inner loop 2:
1 if statement with 2tu operations, and carries out 5tu function and 1tu increment. So, 2+5+1= 8tu

since step of -2 use $\frac{end-start}{step}$ + 1.

8 ($\frac{i-2n-1}{-2}$ + 1) = 8 (n + $\frac{3-i}{2}$)

Combined:
(outer for loop) * (inner loop 1 + inner loop 2)

$\sum_{i=0}^{n-1}$ 2 [8 (n + $\frac{1-i}{2}$) + 8 (n + $\frac{3-i}{2}$)]

= $\sum_{i=0}^{n-1}$ 2 * 8[(n + $\frac{1-i}{2}$) + (n + $\frac{3-i}{2}$)]

= $\sum_{i=0}^{n-1}$ 16 (2n + 2 – i) = $\sum_{i=0}^{n-1}$ 16 (2n – i + 2)

= $\sum_{i=0}^{n-1}(32n - 16i + 32)$

Hammad Qureshi qureshi434@csu.fullerton.edu
Brian Tan brian388@csu.fullerton.edu

= $\sum_{i=0}^{n-1} 16\text{n}$ - $\sum_{i=0}^{n-1} 16i$ + $\sum_{i=0}^{n-1} 32$ // $\sum_{i=0}^{n-1} 16i$ = 16 * 0 + $\sum_{i=1}^{n-1} 16i$ = 16 ($\frac{n-1\,(\,n-1+1\,)}{2}$)

= 16n(n-1 + 1) - 16 ($\frac{n-1\,(\,n-1+1\,)}{2}$) + 32(n-1 + 1)

= $16n^2 - 8n^2 - 8n + 32n$

= $8n^2 - 8n + 32$

Add time units outside of outer loop:

$8n^2 - 8n + 32 + 4 + \frac{5n}{2} + 7 + 8n + 16$

= $8n^2 + \frac{5n}{2} + 59$

highest power is 2. Therefore, algorithm efficiency class is $O(n^2)$.

Proof by definition

Prove $8n^2 + \frac{5n}{2} + 59 \in O(n^2)$

choose $C > 0$ and a value $n_0 >= 0$ such that:

$8n^2 + \frac{5n}{2} + 59 <= C * n^2 \ \forall\ n >= n_0$

choose $C = 8 + \frac{5}{2} + 59 \cong 70$

$8n^2 + \frac{5n}{2} + 59 <= 70n^2$, true for n >= 1

choose $n_0 = 1$

$\therefore\ 8n^2 + \frac{5n}{2} + 59 \in O(n^2)$