Project 1/Lawnmower Project Report

Group Members:

Ranny Khant Naing

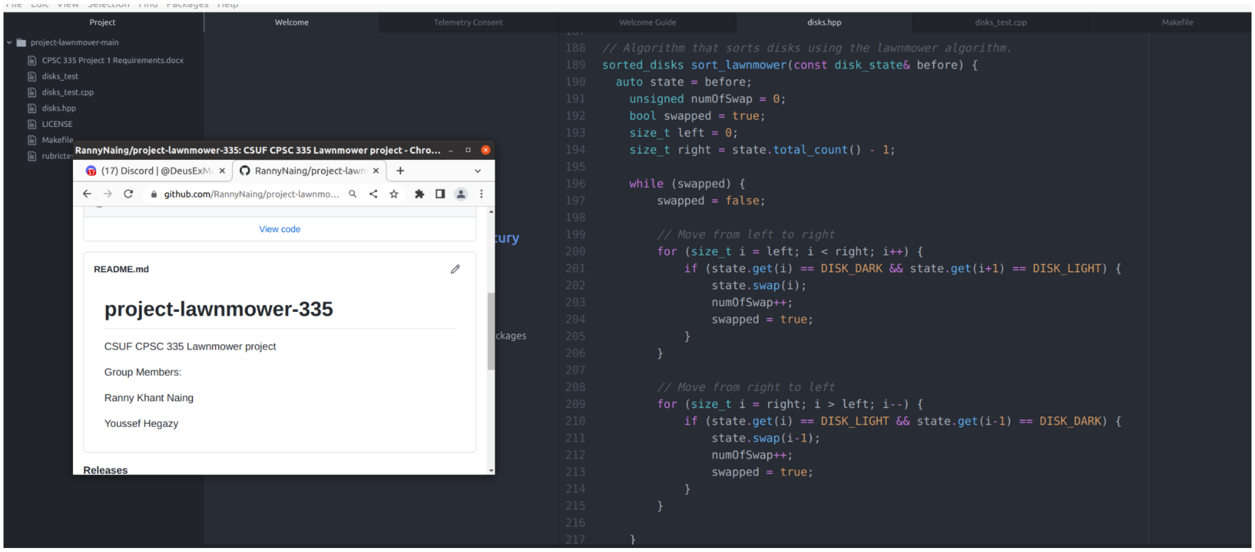
Youssef Hegazy

CSUF Emails:

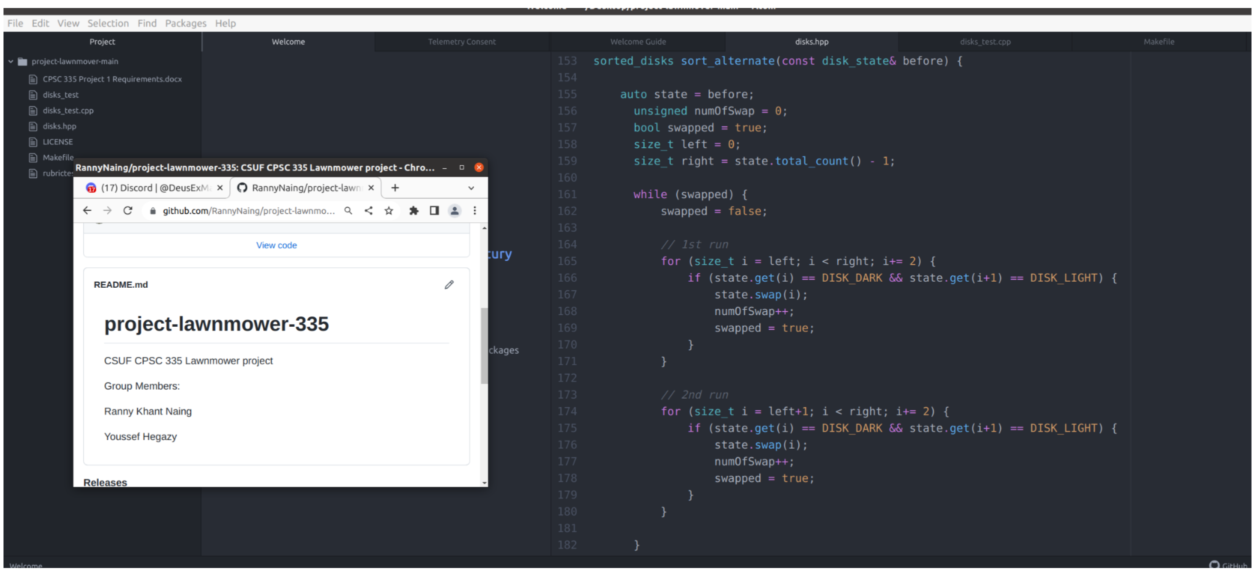
khant_naing@csu.fullerton.edu

YoussefH@csu.fullerton.edu

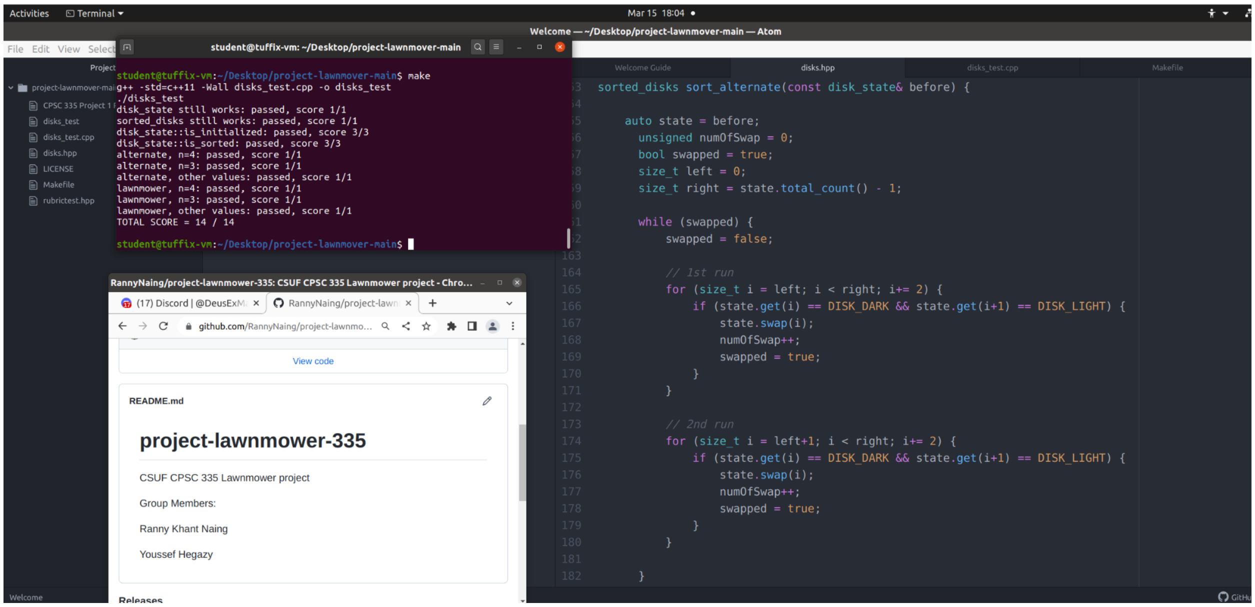
This is a submission for Project 1.



Screenshot for Lawnmower algorithm



Screenshot for Alternate algorithm



Screenshot for the compiling

Pseudo code for Lawnmower algorithm

Def sort_lawnmower(before):	Step Counts
state = before;	1
numOfSwap = 0;	1
swapped = true;	1
left = 0;	1
right = total_count()-1;	2
while (swapped):	n/2

```
        swapped = false;                                        1
        for(i = left; i < right; ++i):                          n
            if(state[i] == dark && state[i+1} == light):        3
                Swap[i];                                        1
                numOfSwap++;                                    1
                swapped = true;                                 1
            end if
        end for
        for(i = right; i > left; --i):                          n
            if(state[i] == light && state[i+1} == dark):        3
                swap[i];                                        1
                numOfSwap++;                                    1
                swapped = true;                                 1
            end if
        end for
    end while
return disk_state(state,numOfSwap);
```

Total Step Count = 1+1+1+1+ 2+ n/2*(1+n*(3+max(3, 0))+n*(3+max(3 0)))

= 6+n/2*(1+6n+6n)

= 6+n/2*(1+12n)

=6n^2 + 6 + n/2

The Lawnmower algorithm has an efficiency of O(n^2).

Time Complexity for Lawnmower algorithm

6n^2 + 6 + n/2 ∈ O(n^2)

By Def

6n^2 + 6 + n/2 ≤ c.(n^2)

choose ,c = 12, n0 = 2

6 * 2^2 + 6 + 2/2 ≤ 12 * 2^2

31 ≤ 48

∴ 6n^2 + 6 + n/2 ∈ O(n^2)

Pseudo code for Alternate algorithm

Def sort_lawnmower(before):	Step Counts
state = before;	1
numOfSwap = 0;	1
swapped = true;	1
left = 0;	1
right = total_count()-1;	2
while (swapped):	n+1
swapped = false;	1

```
        for(i = left; i < right; i += 2):                    n/2
            if(state[i] == dark && state[i+1} == light):     3
                Swap[i];                                     1
                numOfSwap++;                                 1
                swapped = true;                              1
            end if
        end for
        for(i = left +1 ; i < right; i += 2):                n/2
            if(state[i] == dark && state[i+1} == light):     3
                Swap[i];                                     1
                numOfSwap++;                                 1
                swapped = true;                              1
            end if
        end for
    end while
return disk_state(state,numOfSwap);
```

Total Step Count = 1+1+1+1+ 2+ (n+1)*(1+n/2*(3+max(3, 0))+n/2*(3+max(3 0)))

= 6+(n+1)*(1+3n+3n)

= 6+(n+1)*(1+6n)

=6+ n + 6n^2 + 1 + 6n

$=6n^2 + 7n + 7$

The alternate algorithm has an efficiency of $O(n^2)$.

Time Complexity for Alternate algorithm

$6n^2 + 7n + 7 \in O(n^2)$

By Def

$6n^2 + 7n + 7 \leq c.(n^2)$

choose ,$c = 20$, $n0 = 2$

$6 * 2^2 + 7 * 2 + 7 \leq 20 * 2^2$

$45 \leq 80$

$\therefore 6n^2 + 7n + 7 \in O(n^2)$