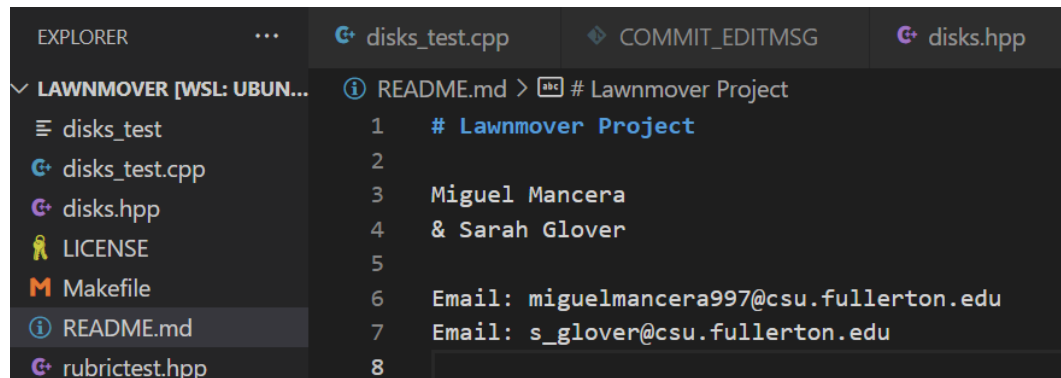


Emails:



The screenshot shows a code editor interface with a dark theme. The Explorer sidebar on the left lists files for a project named 'LAWNMOWER [WSL: UBUN...]', including 'disks_test', 'disks_test.cpp', 'disks.hpp', 'LICENSE', 'Makefile', 'README.md' (selected), and 'rubrictest.hpp'. The main editor area displays the content of 'README.md', which includes a title, authors, and email addresses.

```
1  # Lawnmover Project
2
3  Miguel Mancera
4  & Sarah Glover
5
6  Email: miguelmancera997@csu.fullerton.edu
7  Email: s_glover@csu.fullerton.edu
8
```

Algorithms:

```
151 // Algorithm that sorts disks using the alternate algorithm.
152 sorted_disks sort_alternate(const disk_state& before) {
153     disk_state state = before;
154     int numOfSwap = 0;
155     int iterations = state.light_count()+1;
156
157     for (int i = 0; i < iterations; i++){
158         if ((i%2 == 0)){
159             for (int m = 0; m < int(state.total_count()); m += 2){
160                 if (state.get(m) > state.get(m + 1)){
161                     state.swap(m);
162                     ++numOfSwap;
163                 }
164             }
165         } else {
166             for (int m = 1; m < int(state.total_count()) - 1; m += 2){
167                 if (state.get(m) > state.get(m + 1)){
168                     state.swap(m);
169                     ++numOfSwap;
170                 }
171             }
172         }
173     }
174     return sorted_disks(disk_state(state), numOfSwap);
175 }
```

```
176
177
178 // Algorithm that sorts disks using the lawnmower algorithm.
179 sorted_disks sort_lawnmower(const disk_state& before) {
180     disk_state state = before;
181     int numOfSwap = 0;
182     bool booleanFlag;
183     for (int i = 0; i < state.light_count(); ++i){
184         if (i % 2 == 0) {
185             booleanFlag = true;
186         } else {
187             booleanFlag = false;
188         }
189         for (int m = booleanFlag? 0 : state.total_count() - 2; m < state.total_count() - 1; booleanFlag? ++m : --m){
190             if (state.get(m) > state.get(m + 1)) {
191                 state.swap(m);
192                 ++numOfSwap;
193             }
194         }
195     }
196     return sorted_disks(disk_state(state), numOfSwap);
197 }
```

Execution:

```
● sglover@CSUF-3CN2303:~/Lawnmower$ ./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
```

```
File Edit Selection View Go Run ... disks.hpp - Lawnmower [WSL: Ubuntu] - Visual Studio Code
```

EXPLORER

- disks_test
- disks_test.cpp
- disks.hpp
- LICENSE
- Makefile
- README.md
- rubrictest.hpp

disks.hpp

```
175
176
177
178 // Algorithm that sorts disks using the lawnmower algorithm.
179 sorted_disks sort_lawnmower(const disk_state& before) {
180     disk_state state = before;
181     int numOfSwap = 0;
182     bool booleanFlag;
183     for (int i = 0; i < state.light_count(); ++i){
184         if (i % 2 == 0) {
185             booleanFlag = true;
186         } else {
187             booleanFlag = false;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

bash - Lawnmower

```
./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
sglover@CSUF-3CN2303:~/Lawnmower$
```

WSL: Ubuntu main 0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF C++

Pseudocode:

sort_lawnmower

```
For i = 0 to number_of_iterations-1 do: //Iterations

    For j=0 to 2n-1 do //moving left to right
        if(list[j] == dark AND list[j+1] == light)
            Swap them
    End for

    For j=2n-1 to j=0 do //moving right to left
        if (list[j-1] == dark AND list[j] == light)
            Swap them
    End for

End for
```

Sort_alternative

```
Number_of_iterations = n+1
For i=0 to number_of_iterations-1 do
    If i is even do
        For j=0 to length(list) -1 step 2 do
            If (list[j] == dark AND list[j+1]== light)
                Swap them
        End for
    Else do // if i is odd
        For j=1 to length(list)-2 step 2 do
            if(list[j] == dark AND list[j+1] == light)
                Swap them
        End for
    End for
End for
```

Mathematical Analysis:

Sort Alternative

Step Count & Time Complexity

ITERATIVE ALGORITHM.

$$\begin{aligned} & 1 + 1 + (n/2) \cdot (n-1) \cdot (1 + 1 + 4) \\ &= 2 + (n/2) \cdot (n-1) \cdot (6) \\ &= 2 + 6 \left(\frac{n^2}{2} - \frac{n}{2} \right) \\ &= 6 \frac{n^2}{2} - 6 \frac{n}{2} + 2 \end{aligned}$$

Using limit theorem to prove:

$$6 \frac{n^2}{2} - 6 \frac{n}{2} + 2 \in O(n^2)$$

$$\lim_{n \rightarrow \infty} \frac{6n^2 - 6n + 2}{2n^2}$$

$$= \frac{d}{dn} \frac{12n - 6}{4n} \rightarrow \frac{d}{dn} \frac{12}{4} = 3 \neq \infty$$

Therefore:

$$6 \frac{n^2}{2} - 6 \frac{n}{2} + 2 \in O(n^2)$$

Sort Lawnmower

Step Count:

$$\begin{aligned}
& \frac{n}{2} [2n-1(2+\max(0,1)) + 2n-1(2+\max(0,1))] \\
&= \frac{n}{2} [2n-1(2+1) + 2n-1(2+1)] \\
&= \frac{n}{2} [2n-3 + 2n-3] \\
&= \frac{n}{2} [4n-6] = \frac{4n^2}{2} - \frac{6n}{2} \\
&= \boxed{2n^2 - 3n}
\end{aligned}$$

Time Complexity:

$$2n^2 - 3n \in O(n^2)$$

$$f(n) = 2n^2 - 3n \quad \& \quad g(n) = n^2$$

by def

$$2n^2 - 3n \leq C \cdot n^2$$

$$C = 2 + 3 = 5$$

$$n = 10$$

$$2(10)^2 - 3(10) \leq 5(10)^2$$

$$200 - 30 \leq 500$$

$$170 \leq 500 \quad \checkmark$$

$$\boxed{2n^2 - 3n \in O(n^2)}$$