

Lori Cha

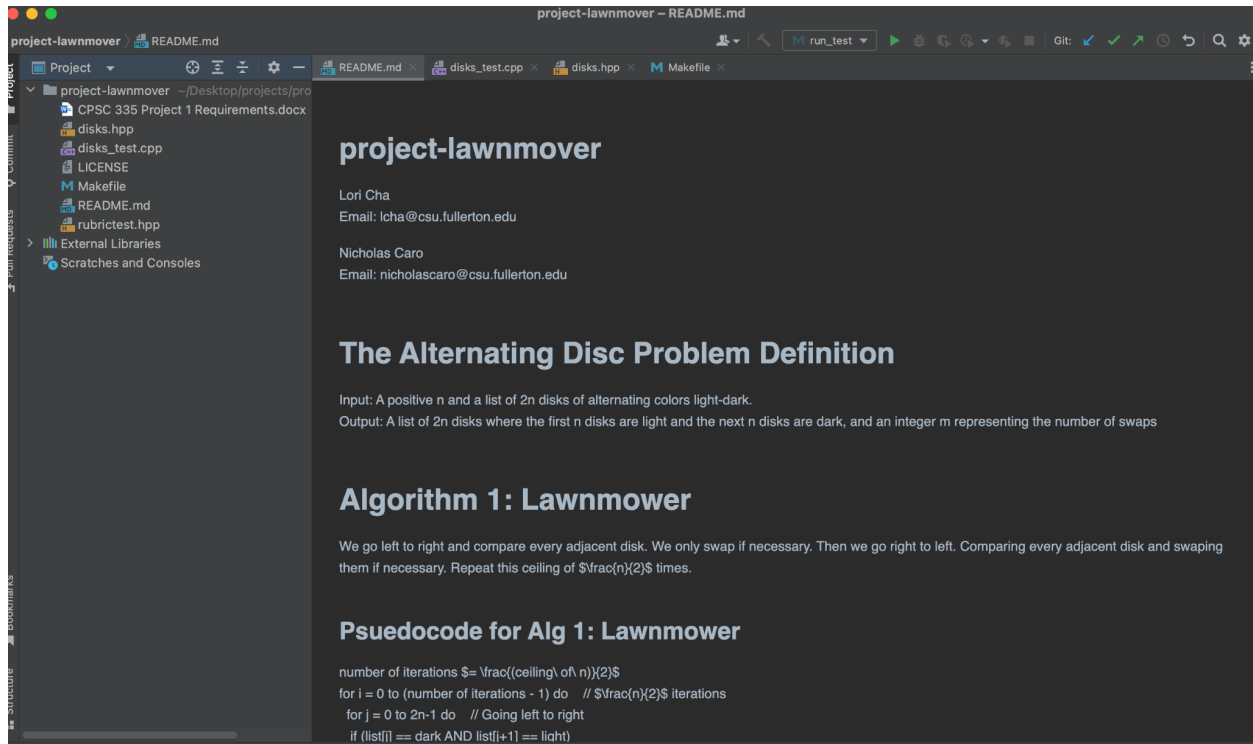
Email: lcha@csu.fullerton.edu

Nicholas Caro

Email: nicholascaro@csu.fullerton.edu

Project 1: Alternating Disks Problem

2. A full-screen screenshot, inside Tuffix, showing the Atom editor or the editor you used, with your group member names shown clearly as below. One way to make your names appear in Atom is to simply open your README.md.



3. A full-screen screenshot showing your code compiling and executing.

```

project-lawnmover - Makefile
Project ~ /Desktop/projects/pro
  add
  CPSC 335 Project 1 Requirements.docx
  disks.hpp
  disks.hpp.gch
  disks_test
  disks_test.cpp
  LICENSE
  Makefile
  README.md
  rubrictest.hpp
  External Libraries
  Scratches and Consoles
6  else
7  CXX_COMMAND := g++
8  endif
9
10 CXX = ${CXX_COMMAND} -std=c++11 -Wall
11
12 run_test: disks_test
13     ./disks_test
14
15 headers: rubrictest.hpp disks.hpp
16
17 disks_test: headers disks_test.cpp
18     ${CXX} disks_test.cpp -o disks_test
19
Run: M run_test
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
Process finished with exit code 0

```

4. Two Pseudocode Listings, for the two algorithms and their step count

Pseudocode for Alg 1: Lawnmower

number of iterations = $\frac{\text{ceiling of } n}{2}$

for i = 0 to (number of iterations - 1) do // $\frac{n}{2}$ iterations

 for j = 0 to 2n-1 do // Going left to right

 if (list[j] == dark AND list[j+1] == light)

 Swap them

 end for

 for j = 2n-1 to j = 0 do // Going back right to left

 if (list[j-1] == dark AND list[j] == light)

 Swap them

 end for

end for

Step Count

$$\begin{aligned}
 S.C. &= \frac{n}{2} (2n - 1(2 + \max(0, 1)) + 2n - 1(2 + \max(0, 1))) \\
 &= \frac{n}{2} (2n - 1(2 + 1)) + 2n - 1(2 + 1) \\
 &= \frac{n}{2} (2n - 1(3)) + 2n - 1(3) \\
 &= \frac{n}{2} ((6n - 3) + (6n - 3)) \\
 &= \frac{n}{2} (12n - 6) \\
 &= 6n^2 - 3n
 \end{aligned}$$

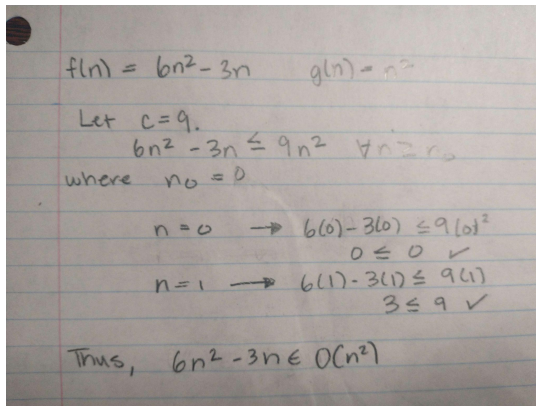
Psuedocode for Alg 2

```
number of iterations = n + 1
for i = 0 to (# 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
    end do
  else do
    for j = 1 to length(list) - 2 step 2 do
      if (list[j] == dark AND list[j+1] == light)
        Swap them
      end for
    end do
  end for
end for
```

Step Count

$$\begin{aligned} S.C. &= n(1 + \max(\frac{n-1}{2}(2+1), \frac{n-2}{2}(2+1))) \\ &= n(1 + (\frac{n-1}{2}(3))) \\ &= n(1 + (\frac{3}{2}(n-1))) \\ &= n + \frac{3}{2}n^2 - \frac{3}{2}n \\ &= \frac{3}{2}n^2 - \frac{1}{2}n \end{aligned}$$

5. A brief proof argument for the time complexity of your two algorithms
For Alg 1:



$f(n) = 6n^2 - 3n$ $g(n) = n^2$

Let $c=9$.

$6n^2 - 3n \leq 9n^2 \quad \forall n \geq n_0$

where $n_0 = 0$

$n=0 \rightarrow 6(0) - 3(0) \leq 9(0)^2$
 $0 \leq 0 \quad \checkmark$

$n=1 \rightarrow 6(1) - 3(1) \leq 9(1)$
 $3 \leq 9 \quad \checkmark$

Thus, $6n^2 - 3n \in O(n^2)$

Efficiency Class Using Limit Theorem

$$\lim_{n \rightarrow \infty} \frac{6n^2 - 3n}{n^2} = 6$$

Since $6 \geq 0$ and 6 is a constant, the Limit Theorem tells us that $6n^2 - 3n \in O(n^2)$.

Thus, this implementation of this algorithm is $O(n^2)$.

For Alg 2:

$$f(n) = \frac{3}{2}n^2 - \frac{1}{2}n \quad g(n) = n^2$$

Let $c = 2$ and $n_0 = 2$.

Notice, $\frac{3}{2}n^2 - \frac{1}{2}n \leq 2n^2 \quad \forall n \geq n_0$

$n = 0 \rightarrow \frac{3}{2}(0) - \frac{1}{2}(0) \leq 2(0)^2$
 $0 \leq 0$

$n = 1 \rightarrow \frac{3}{2}(1)^2 - \frac{1}{2}(1) \leq 2(1)^2$
 $1 \leq 2 \quad \checkmark$

Thus, $\frac{3}{2}n^2 - \frac{1}{2}n \in O(n^2)$

Efficiency Class Using Limit Theorem

$$\lim_{n \rightarrow \infty} \frac{\frac{3}{2}n^2 - \frac{1}{2}n}{n^2} = \frac{3}{2}$$

Since $\frac{3}{2} \geq 0$ and $\frac{3}{2}$ is a constant, the Limit Theorem tells us that $\frac{3}{2}n^2 - \frac{1}{2}n \in O(n^2)$

Thus, this implementation of this algorithm is $O(n^2)$.