

Table of Contents

Group members	2
Screenshots of Editor and Code Compiling/Executing	3
<i>Editor</i>	3
<i>Code Compiling and Executing</i>	4
Alternate Algorithm	5
<i>Pseudocode</i>	5
<i>Step Count</i>	6
<i>Efficiency Class with Limit Theorem</i>	7
Lawnmower Algorithm	8
<i>Pseudocode</i>	8
<i>Step Count</i>	9
<i>Efficiency Class with Limit Theorem</i>	10

Group members

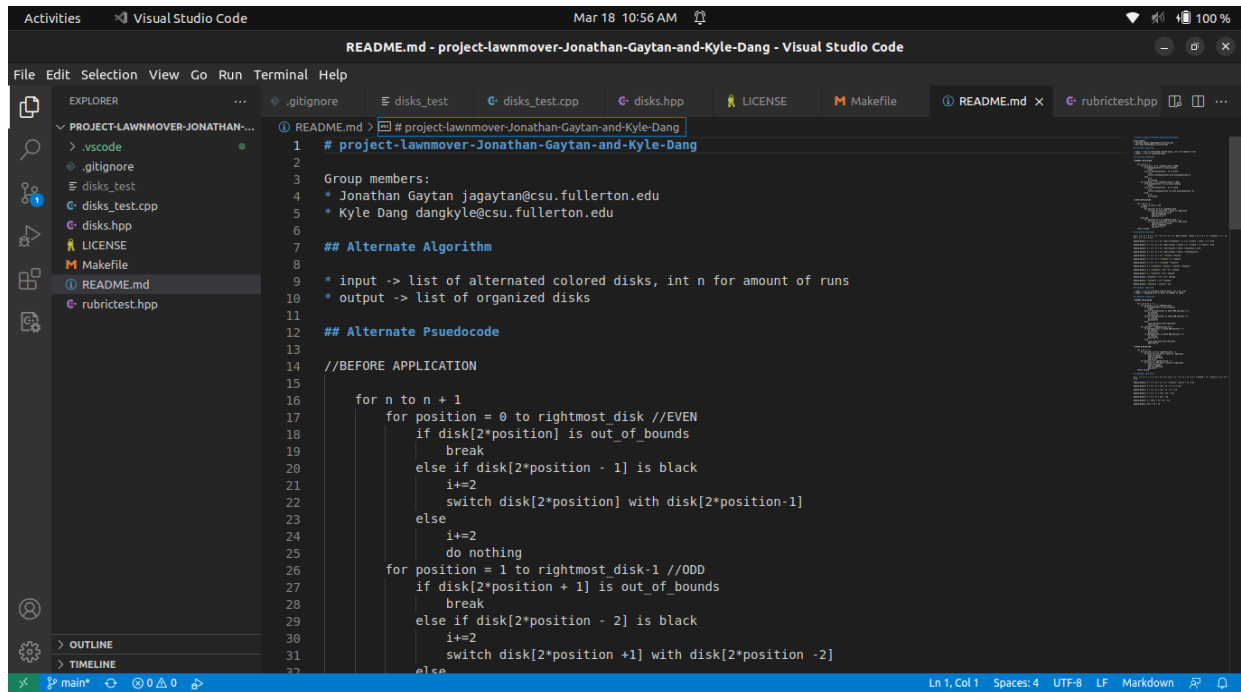
Jonathan Gaytan jagaytan@csu.fullerton.edu

Kyle Dang dangkyle@csu.fullerton.edu

Submission for Project 1

Screenshots of Editor and Code Compiling/Executing

Editor



Code Compiling and Executing

Visual Studio Code interface showing the compilation and execution of a C++ program. The editor displays the source code for `disks.hpp` and `disks_test.cpp`. The terminal shows the output of the program, including test results and a final score of 14/14.

```

disks.hpp - project-lawnmower-Jonathan-Gaytan-and-Kyle-Dang - Visual Studio Code

File Edit Selection View Go Run Terminal Help

RUN AND DEBUG
Run and Debug
To customize Run and Debug create a launch.json file.
Show all automatic debug configurations.
To learn more about launch.json, see Configuring C/C++ debugging.

disks.hpp x .gitignore disks_test.cpp

disks.hpp > sort_lawnmower(const disk_state &)
// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {
    disk_state state = before;
    int numOfSwap = 0;
    int light_total = state.light_count() + 1;
    int rightmost_disk = state.total_count();

    for (int n = 0; n < light_total; n++) {
        if (n % 2 == 0) {
            for (int position = 0; position < rightmost_disk; position += 2) {

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

[1] + Done "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-3vc
jrd5r.lmc" 1>"/tmp/Microsoft-MIEngine-Out-m2kdyabz.yqf"
darkmatter@darkmatter-HP-Pavilion-Laptop-15-cx1xx:~/CPSC-335-Algorithm-Engineering/project-lawnmower-Jonathan-Gaytan-and-Kyle-Dang$
  
```

Alternate Algorithm

Pseudocode

```
for n to n + 1
  if even
    for position at 0 to rightmost_disk
      if value of left_disk > value of right_disk
        swap position of disks
        add to numOfSwap
        position += 2
  else odd
    for position at 1 to rightmost disk - 1
      if value of left_disk > value of right_disk
        swap position of disks
        add to numOfSwap
        position += 2
return sorted
```

Step Count

$$\begin{aligned}
s.c. &= 1 + 1 + 2 + 1 + (n - 0 + 1) * (2 + 2 + \max((\frac{n-0}{2} + 1) * (2 + 1), ((\frac{(n-1)-1}{2} + 1) * (2 + 1)))) \\
&= 5 + (n + 1) * (4 + \max((\frac{n}{2} + 1) * 3, (\frac{n-2}{2} + 1) * 3)) \\
&= 5 + (n + 1) * (4 + \max((\frac{n+2}{2}) * 3, (\frac{n-2+2}{2}) * 3)) \\
&= 5 + (n + 1) * (4 + \max(\frac{3n+6}{2}, \frac{n}{2} * 3)) \\
&= 5 + (n + 1) * (4 + \max(\frac{3n+6}{2}, \frac{3n}{2})) \\
&= 5 + (n + 1) * (4 + \frac{3n+6}{2}) \\
&= 5 + (n + 1) * (\frac{3n+6+8}{2}) \\
&= 5 + (n + 1) * (\frac{3n+14}{2}) \\
&= 5 + (\frac{3n^2+14n}{2} + \frac{3n+14}{2}) \\
&= 5 + (\frac{3n^2+14n+3n+14}{2}) \\
&= 5 + (\frac{3n^2+17n+14}{2}) \\
&= \frac{3n^2+17n+14+10}{2} \\
&= \frac{3n^2+17n+24}{2} \\
&= \frac{3n^2+17n}{2} + 12
\end{aligned}$$

Efficiency Class with Limit Theorem

$$\lim_{n \rightarrow \infty} \frac{\frac{3}{2}n^2 + \frac{17}{2}n + 12}{n^2} = \frac{3}{2}$$

Due to $\frac{3}{2} \geq 0$ and $\frac{3}{2}$ being a constant, the Limit Theorem states that $\frac{3n^2 + 17n}{2} + 12 \in O(n^2)$.

That means this algorithm has a time complexity of $O(n^2)$.

Lawnmower Algorithm

Pseudocode

```
for n to n / 2
  for position at 0 to rightmost_disk - 1
    if value of left_disk > value of right_disk
      swap the disks
      add to numOfSwap
      position++
  for position at rightmost_disk - 2
    if value of left_disk > value of right_disk
      swap the disks
      add to numOfSwap
      position--
return sorted
```


Step Count

$$\begin{aligned}
s.c. &= 1 + 1 + 1 + 1 + (n - 0 + 1) * (((n - 1) - 0 + 1) * (2 + 1) + (\frac{0-(n-2)}{-1} + 1) * (2 + 1)) \\
&= 4 + (n + 1) * (n * 3 + (\frac{-n+2}{-1} + 1) * 3) \\
&= 4 + (n + 1) * (3n + (n - 2 + 1) * 3) \\
&= 4 + (n + 1) * (3n + (n - 1) * 3) \\
&= 4 + (n + 1) * (3n + (3n - 3)) \\
&= 4 + (n + 1) * (6n - 3) \\
&= 4 + (6n^2 - 3n + 6n - 3) \\
&= 6n^2 + 3n + 1
\end{aligned}$$

Efficiency Class with Limit Theorem

$$\lim_{n \rightarrow \infty} \frac{6n^2 + 3n + 1}{n^2} = 6$$

Due to $6 \geq 0$ and 6 being a constant, the Limit Theorem states that $6n^2 + 3n + 1 \in O(n^2)$.

That means this algorithm has a time complexity of $O(n^2)$.