

**Input:** a positive integer  $n$  and a list of  $2n$  disks of alternating colors light-dark, starting with light

**Output:** a list of  $2n$  disks, the first  $n$  disks are light, the next  $n$  disks are dark, and an integer  $m$  representing the number of swaps to move the dark ones after the light ones.

Algorithm 1 - Lawnmower Algorithm:

### Algorithm Design:

**Input:** List of length  $2n$  with disks of alternating colors. (Assume light is less than dark)

**Pseudocode:**

first = 0

for  $j = 0$  to  $n/2$  do

    //loop forward, first check

    for  $i = 0$  to  $(2n - 1)$  do

        //compare forward

        if  $\text{list}[i] < \text{list}[i+1]$

            first =  $\text{list}[i]$

$\text{list}[i] = \text{list}[i+1]$

$\text{list}[i+1] = \text{first}$

        end if

    end for

    //finished first loop, now to loop backwards from  $2n$

    for  $i = (2n - 1)$  to  $0$  do

        //compare backwards

        if  $\text{list}[i] > \text{list}[i-1]$

            first =  $\text{list}[i]$

$\text{list}[i] = \text{list}[i-1]$

$\text{list}[i-1] = \text{first}$

        end if

    end for

    //finished second loop, move onto next iteration of master loop, or finish

end for

## Mathematical Analysis:

Step Count:

$$\text{Step Count} = 2 + \lfloor n/2 \rfloor * (\text{SC1} + \text{SC2})$$

$$\text{SC1} = [2n - 1 + 1] * (2 + 1 + 2 + 2)$$

$$\rightarrow = [2n] * (7)$$

$$\rightarrow = 14n$$

$$\text{SC2} = [ ((0 - (2n-1)) / (-1)) + 1 ] * (2 + 1 + 2 + 2)$$

$$\rightarrow = [2n] * (7)$$

$$\rightarrow = 14n$$

$$\text{Step Count} = 2 + \lfloor n/2 \rfloor * (14n + 14n)$$

$$\rightarrow = 2 + \lfloor n/2 \rfloor * (28n)$$

$$\rightarrow = 2 + (14n^2)$$

$$\rightarrow = 14n^2 + 1$$

$$\text{Final Step Count} = 14n^2 + 1$$

Big O Notation:

Suppose  $F(n) = 14n^2 + 1$  and  $G(n) = n^2$

We want to show that there exists some  $C$  and some  $N_0$  such that

$$F(n) < C * G(n) \text{ for } n > N_0$$

Let  $C = 100$  and  $N_0 = 100$

$$F(n) < 100 * G(n) \text{ for } n > 100$$

$$\rightarrow 14n^2 + 1 < 100 * n^2 \text{ for } n > 100$$

$$\rightarrow 14 * (100)^2 + 1 < 100 * (100)^2$$

$$\rightarrow 14 * (10,000) + 1 < 100 * (10,000)$$

$$\rightarrow 140,000 + 1 < 1,000,000$$

$$\rightarrow 140,001 < 1,000,000$$

Therefore, because  $F(n) < 100 * G(n)$  for  $n > 100$ , we can conclude that by the definition of Big O Notation that  $14n^2 + n/2 + 1$  exists within  $O(n^2)$ .

Algorithm 2 - Alternate Algorithm:

### **Algorithm Design:**

**Input:** List of length  $2n$  with disks of alternating colors. (Assume light is less than dark)

**Pseudocode:**

```
for j = 0 to n
    for i = 0 to 2n, step 2 do
        if list[i] < list[i+1]
            first = list[i]
            list[i] = list[i+1]
            list[i+1] = first
        end if
    end for

    for i = 1 to 2n - 1, step 2 do
        if list[i] < list[i+1]
            first = list[i]
            list[i] = list[i+1]
            list[i+1] = first
        end if
    end for
end for
```

### **Mathematical Analysis:**

lmao

Step Count:

Step Count =  $[n + 1] * (SC1 + SC2)$

$$SC1 = [(2n)/2 + 1] * (2 + 1 + 2 + 2)$$

$$\rightarrow [n + 1] * (7)$$

$$\rightarrow 7n + 7$$

$$SC2 = [ ((2n-1) - 1)/2 + 1] * (2 + 1 + 2 + 2)$$

$$\rightarrow [(2n-2)/2 + 1] * (7)$$

$$\rightarrow [n] * (7)$$

$$\rightarrow 7n$$

$$\text{Step Count} = [n+1] * ((7n + 7) + (7n))$$

$$\rightarrow [n+1] * (14n + 7)$$

$$\rightarrow 14n^2 + 7n + 14n + 7$$

$$\rightarrow 14n^2 + 21n + 7$$

$$\text{Final Step Count} = 14n^2 + 21n + 7$$

Big O Notation:

Suppose  $F(n) = 14n^2 + 21n + 7$  and  $G(n) = n^2$

We want to show that there exists some C and some  $N_0$  such that

$$F(n) < C * G(n) \text{ for } n > N_0$$

Let  $C = 100$  and  $N_0 = 100$

$$F(n) < C * G(n) \text{ for } n > N_0$$

$$\rightarrow F(100) < 100 * G(100)$$

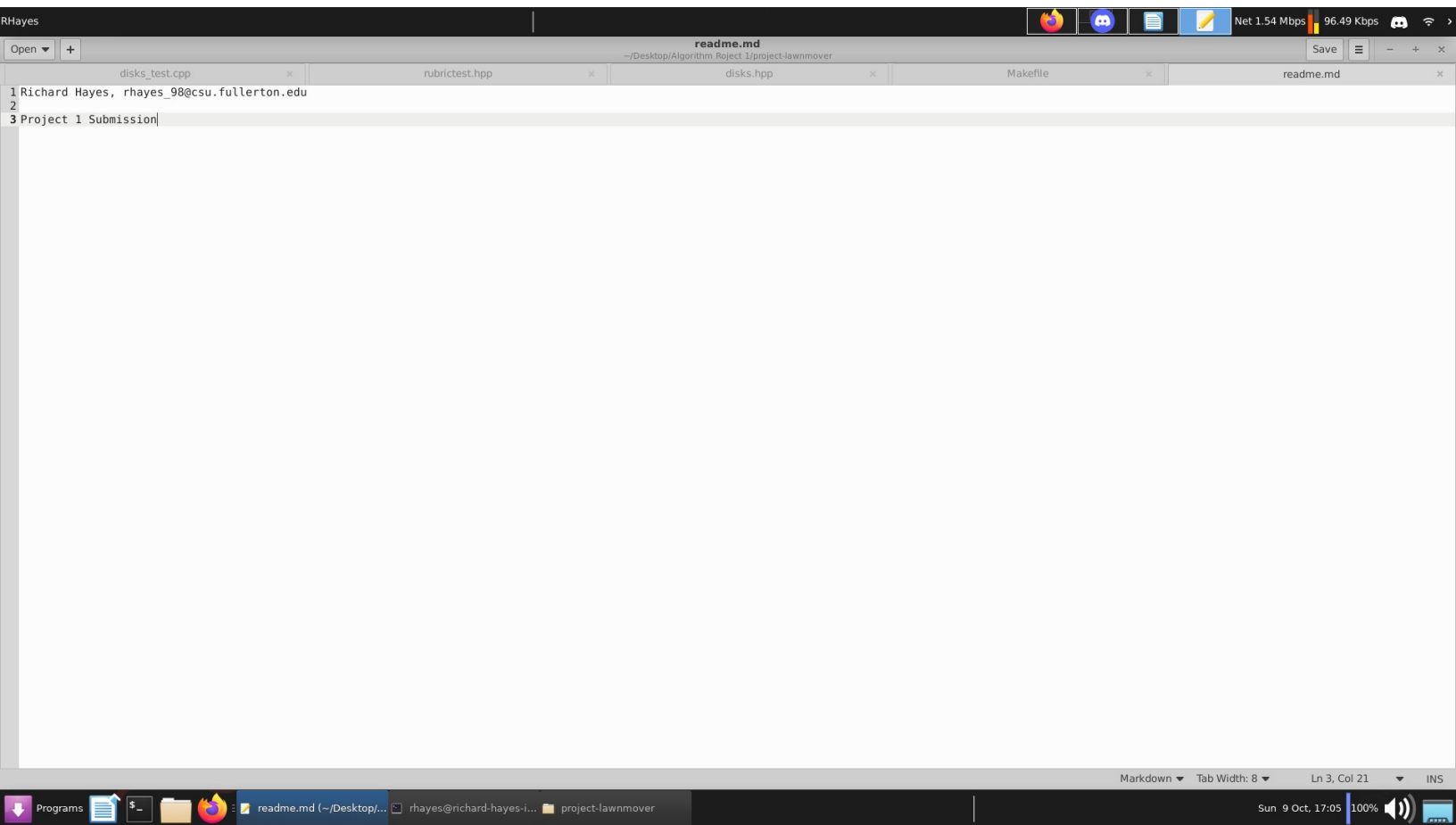
$$\rightarrow 14(100)^2 + 21(100) + 7 < 100 (100)^2$$

$$\rightarrow 140,000 + 2100 + 7 < 1,000,000$$

$$\rightarrow 142,107 < 1,000,000 \text{ is true}$$

Therefore, because  $F(n) < 100 * G(n)$  for  $n > 100$ , we can conclude that by the definition of Big O Notation that  $14n^2 + 21n + 7$  exists within  $O(n^2)$ .

Readme.md Screenshot:



## Code running Screenshot:

The screenshot shows a C++ development environment. The terminal window displays the output of a `make` command, which runs a test suite for a program called `lawnmover`. The test results show several failures, including "TEST FAILED: line 89 of file disks\_test.cpp, message: actually sorted" and "TEST FAILED: line 107 of file disks\_test.cpp, message: n=10 gives 45 swaps". The total score is 8 / 14.

```
rhayes@richard-hayes-inspiron: ~/Desktop/Algorithm Roject 1/project-lawnmover
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:
TEST FAILED:
line 89 of file disks_test.cpp, message: actually sorted
score 0/1
alternate, n=3:
TEST FAILED:
line 96 of file disks_test.cpp, message: actually sorted
score 0/1
alternate, other values:
TEST FAILED:
line 107 of file disks_test.cpp, message: n=10 gives 45 swaps
score 0/1
lawnmower, n=4:
TEST FAILED:
line 122 of file disks_test.cpp, message: actually sorted
score 0/1
lawnmower, n=3:
TEST FAILED:
line 129 of file disks_test.cpp, message: actually sorted
score 0/1
lawnmower, other values:
TEST FAILED:
line 140 of file disks_test.cpp, message: n=10 gives 45 swaps
score 0/1
TOTAL SCORE = 8 / 14
make: *** [Makefile:13: run_test] Error 1
rhayes@richard-hayes-inspiron:~/Desktop/Algorithm Roject 1/project-lawnmover$
```

The code editor shows a `Makefile` with the following content:

```
Makefile
# C++ Compiler
CXX = g++
# Compiler flags
CXXFLAGS = -std=c++11 -Wall
# Source files
SRCS = disks.cpp
# Object files
OBJS = $(SRCS:.cpp=.o)
# Executable
EXEC = lawn_mover
# Linker flags
LDFLAGS =
# Default target
all: $(EXEC)
$(EXEC): $(OBJS)
$(CXX) $(CXXFLAGS) $(SRCS) $(LDFLAGS) -o $(EXEC)
%.o: %.cpp
$(CXX) $(CXXFLAGS) -c $< -o $>
# Test target
test: $(EXEC)
$(EXEC) ./disks_test
# Clean target
clean:
rm -f $(OBJS) $(EXEC)
```