

Peining Lo

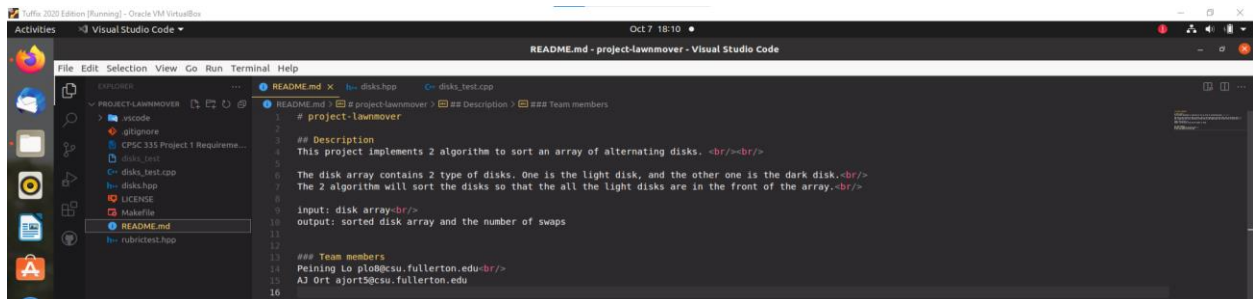
AJ Ort

Plo8@csu.fullerton.edu

Ajort5@csu.fullerton.edu

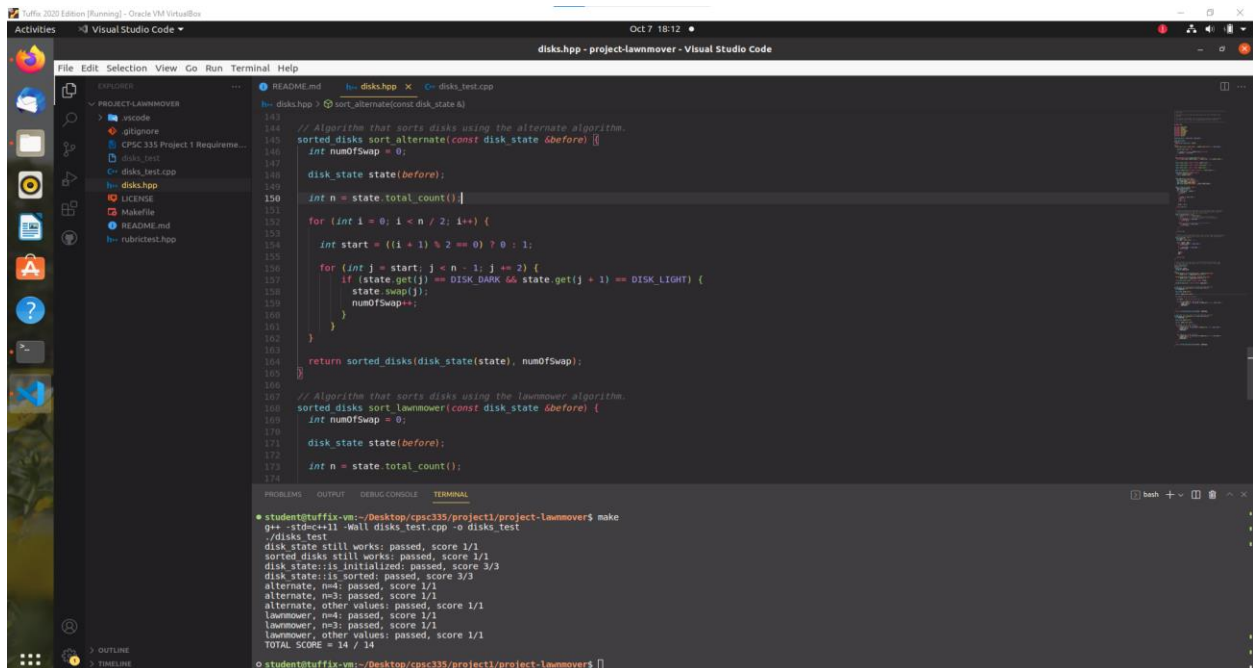
CPSC 335-02 13561

Project 1



The screenshot shows the Visual Studio Code editor with the README.md file open. The file contains the following content:

```
1 # project-lawnmower
2
3 ## Description
4 This project implements 2 algorithms to sort an array of alternating disks. <br/>=<br/>
5 The disk array contains 2 type of disks. One is the light disk, and the other one is the dark disk.<br/>
6 The 2 algorithm will sort the disks so that the all the light disks are in the front of the array.<br/>
7
8 Input: disk array<br/>
9 output: sorted disk array and the number of swaps
10
11
12 ## Team members
13 Peining Lo plo8@csu.fullerton.edu<br/>
14 AJ Ort ajort5@csu.fullerton.edu
15
16
```



The screenshot shows the Visual Studio Code editor with the disks.hpp file open. The file contains the following C++ code:

```
144 // Algorithm that sorts disks using the alternate algorithm.
145 sorted_disks sort_alternate(const disk_state &before) {
146     int numOfSwaps = 0;
147     disk_state state(before);
148
149     int n = state.total_count();
150     for (int i = 0; i < n / 2; i++) {
151         int start = ((i + 1) % 2 == 0) ? 0 : 1;
152         for (int j = start; j < n - 1; j += 2) {
153             if (state.get(j) == DISK_DARK && state.get(j + 1) == DISK_LIGHT) {
154                 state.swap(j);
155                 numOfSwaps++;
156             }
157         }
158     }
159     return sorted_disks(disk_state(state), numOfSwaps);
160 }
161
162 // Algorithm that sorts disks using the lawnmower algorithm.
163 sorted_disks sort_lawnmower(const disk_state &before) {
164     int numOfSwaps = 0;
165     disk_state state(before);
166     int n = state.total_count();
167
168     for (int i = 0; i < n - 1; i++) {
169         if (state.get(i) == DISK_DARK && state.get(i + 1) == DISK_LIGHT) {
170             state.swap(i);
171             numOfSwaps++;
172         }
173     }
174 }
```

The terminal output shows the following results:

```
student@tuffix-vm:~/Desktop/cpsc335/project1/project-lawnmower$ make
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
```

Pseudocode

Lawnmower

```
function lawnmower(diskArr) {  
    numSwap = 0  
    newDiskArr = diskArr  
    n = total number of disks  
  
    for(i = 0 to n/4 - 1) do  
        for(j = 0 to n-2) do  
            if(newDiskArr[j] is dark AND newDiskArr[j+1] is light)  
                swap newDiskArr[j] and newDiskArr[j+1]  
                numSwap++  
            endif  
        endfor  
        for(j = n-1 to 1) step down, do  
            if(newDiskArr[j] is light AND newDiskArr[j-1] is dark)  
                swap newDiskArr[j] and newDiskArr[j-1]  
                numSwap++  
            endif  
        endfor  
    endfor  
  
    return (newDiskArr, numSwap)  
}
```

Alternate

```
function alternate(diskArr) {  
    numSwap = 0  
    newDiskArr = diskArr  
    n = total number of disks  
  
    for(i = 0 to n/2 - 1)  
        start = 0  
        if( (i+1) % 2 == 0)  
            start = 0  
        endif  
    else  
        start = 1  
    endelse  
  
    for(j = start to n-2) step 2  
        if(newDiskArr[j] is dark AND newDiskArr[j+1] is light  
            swap newDiskArr[j] and newDiskArr[j+1]  
            numSwap++  
        endif  
    endfor  
endfor  
  
    return (newDiskArr, numSwap)  
}
```

Step Count

Lawnmower

```
function lawnmower(diskArr) {  
    numSwap = 0 1  
    newDiskArr = diskArr 1  
    n = total number of disks 1  
  
    for(i = 0 to n/4 - 1) do  
        for(j = 0 to n-2) do  
            if(newDiskArr[j] is dark AND newDiskArr[j+1] is light) 5  
                swap newDiskArr[j] and newDiskArr[j+1] 1  
                numSwap++ 1  
            endif  
        endfor  
        for(j = n-1 to 1) step down, do  
            if(newDiskArr[j] is light AND newDiskArr[j-1] is dark) 5  
                swap newDiskArr[j] and newDiskArr[j-1] 1  
                numSwap++ 1  
            endif  
        endfor  
    endfor  
  
    return (newDiskArr, numSwap)  
}
```

for(j = 0 to n-2),

$$sc = \sum_0^{n-2} 7 = \sum_1^{n-1} 7 = 7n - 7$$

for(j = n-1 to 1) step down

$$sc = \sum_1^{n-1} 7 = 7n - 7$$

$$7n - 7 + (7n - 7) = 14n - 14$$

for(i = 0 to n/4 - 1)

$$sc = \sum_0^{\frac{n}{4}-1} (14n - 14) = \sum_1^{\frac{n}{4}} (14n - 14) = (14n - 14) \frac{n}{4} = \frac{7}{2}n^2 - \frac{7}{2}n$$

$$Total\ sc = \frac{7}{2}n^2 - \frac{7}{2}n + 3$$

Alternate

```
function alternate(diskArr) {  
    numSwap = 0                                1  
    newDiskArr = diskArr                      1  
    n = total number of disks                 1  
  
    for(i = 0 to n/2 - 1)  
        start = 0                               1  
        if( (i+1) % 2 == 0)                     3  
            start = 0                             1  
        endif  
    else  
        start = 1                               1  
    endelse  
  
    for(j = start to n-2) step 2  
        if(newDiskArr[j] is dark AND newDiskArr[j+1] is light 5  
            swap newDiskArr[j] and newDiskArr[j+1]          1  
            numSwap++                                         1  
        endif  
    endfor  
endfor  
  
    return (newDiskArr, numSwap)  
}
```

for(j = start to n-2) step 2

in this case, start can only be 0 OR 1, use 0 because we are taking the upper bound

$$\sum_0^{n-2} 7 \text{ step2} = \sum_1^{n-1} 7 \text{ step2} = \sum_1^{\frac{n-1}{2}} 7 = \frac{7}{2}n - \frac{7}{2}$$

$$\frac{7}{2}n - \frac{7}{2} + 5 = \frac{7}{2}n + \frac{3}{2}$$

for(i = 0 to n/2 - 1)

$$\sum_0^{\frac{n}{2}-1} \left(\frac{7}{2}n + \frac{3}{2}\right) = \sum_1^{\frac{n}{2}} \left(\frac{7}{2}n + \frac{3}{2}\right) = \frac{7}{2}n + \frac{3}{2} \left(\frac{n}{2}\right) = \frac{7}{4}n^2 + \frac{3}{4}n$$

$$Total\ sc = \frac{7}{4}n^2 + \frac{3}{4}n + 3$$

Analysis

Lawnmower

$$Total\ sc = \frac{7}{2}n^2 - \frac{7}{2}n + 3 \in O(n^2)$$

Proof by definition:

$$f(n) = \frac{7}{2}n^2 - \frac{7}{2}n + 3$$

$$g(n) = n^2$$

$$f(n) \leq c * g(n) \text{ for } n \geq n_0$$

$$c = \frac{7}{2} + \frac{7}{2} + 3 = 10$$

$$n_0 = 1$$

$$\frac{7}{2} - \frac{7}{2} + 3 = 3 \leq 10$$

Proof by limit theorem:

$$L = \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{\frac{7}{2}n^2 - \frac{7}{2}n + 3}{n^2} = \lim_{n \rightarrow \infty} \frac{(\frac{7}{2}n^2 - \frac{7}{2}n + 3)'}{(n^2)'} = \lim_{n \rightarrow \infty} \frac{\frac{7}{2}n - \frac{7}{2}}{n} = \lim_{n \rightarrow \infty} \frac{\frac{7}{2}}{1} = \frac{7}{2}$$

$$L = \frac{7}{2} \geq 0 \text{ and is a constant}$$

Alternate

$$Total\ sc = \frac{7}{4}n^2 + \frac{3}{4}n + 3$$

Proof by definition:

$$f(n) = \frac{7}{4}n^2 + \frac{3}{4}n + 3$$

$$g(n) = n^2$$

$$f(n) \leq c * g(n) \text{ for } n \geq n_0$$

$$c = \frac{7}{4} + \frac{3}{4} + 3 = \frac{11}{2}$$

$$n_0 = 1$$

$$\frac{7}{4} + \frac{3}{4} + 3 = \frac{11}{2} \leq \frac{11}{2}$$

Proof by limit theorem:

$$L = \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \lim_{n \rightarrow \infty} \frac{\frac{7}{4}n^2 + \frac{3}{4}n + 3}{n^2} = \lim_{n \rightarrow \infty} \frac{(\frac{7}{4}n^2 + \frac{3}{4}n + 3)'}{(n^2)'} = \lim_{n \rightarrow \infty} \frac{\frac{7}{2}n + \frac{7}{4}}{2n} = \lim_{n \rightarrow \infty} \frac{\frac{7}{2}}{2} = \frac{7}{4}$$

$$L = \frac{7}{4} \geq 0 \text{ and is a constant}$$