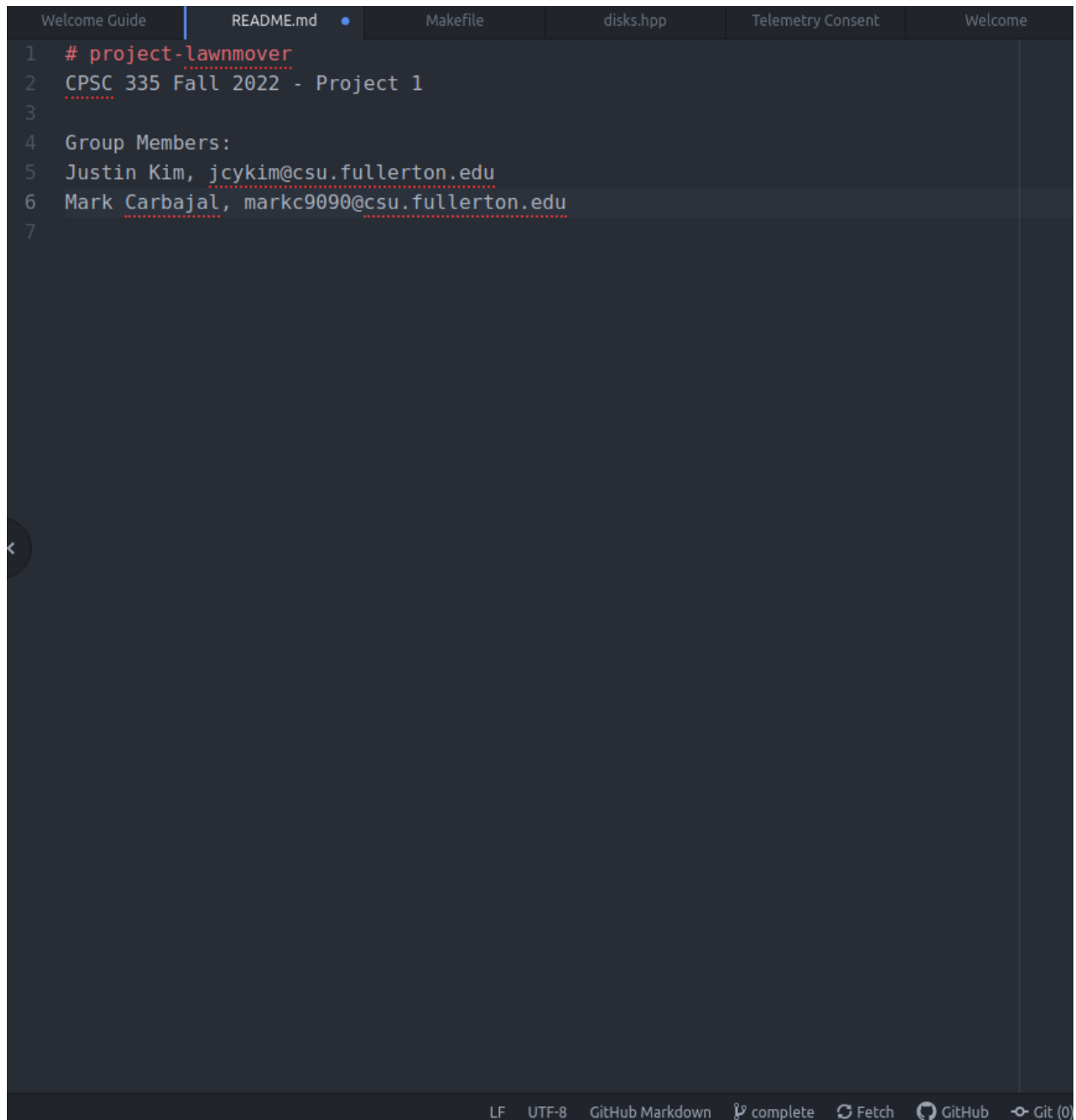


Justin Kim - jcykim@csu.fullerton.edu
Mark Carbajal - markc9090@csu.fullerton.edu

CPSC 335 Project 1

Due: 10/9/2022



The image shows a screenshot of a code editor interface. At the top, there is a tab bar with several tabs: 'Welcome Guide', 'README.md' (which is active and has a blue dot), 'Makefile', 'disks.hpp', 'Telemetry Consent', and 'Welcome'. The main area of the editor displays the content of the README.md file. The text is as follows:

```
1 # project-lawnmover
2 CPSC 335 Fall 2022 - Project 1
3
4 Group Members:
5 Justin Kim, jcykim@csu.fullerton.edu
6 Mark Carbajal, markc9090@csu.fullerton.edu
7
```

At the bottom of the editor, there is a status bar with the following information: 'LF', 'UTF-8', 'GitHub Markdown', 'complete', 'Fetch', 'GitHub', and 'Git (0)'.

The screenshot shows the Visual Studio Code editor with the 'disks.hpp' file open. The file contains C++ code for a disk sorting algorithm. The terminal window at the bottom shows the output of a 'make' command, displaying test results for various disk states and configurations.

```
disks.hpp
147 };
148
149 // Algorithm that sorts disks using the alternate algorithm.
150 sorted_disks sort_alternate(const disk_state& before) {
151
152     int numOfSwap = 0; //record # of step swap
153     disk_state state(before);
154     int n = state.light_count();
155
156     for(int k = 0; k < n+1; ++k ){
157         for(int i = 0; i < n*2; i+=2 ){
158             if( state.get(i) == DISK_DARK && state.get(i+1) == DISK_LIGHT){
159                 state.swap(i);
160                 ++numOfSwap;
161             }
162         }
163     }
164
165     for(int i = 1; i < n*2; i+=2 ){
166         if( state.get(i) == DISK_DARK && state.get(i+1) == DISK_LIGHT){
167             state.swap(i);
168             ++numOfSwap;
169         }
170     }
171
172     return state;
173 }
```

```
justin@justin-MacBookPro:~/Documents/CSUF/Fall2022/CPSC335/project-lawnmover$ make -f Makefile
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
justin@justin-MacBookPro:~/Documents/CSUF/Fall2022/CPSC335/project-lawnmover$
```

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.

Lawnmower Algorithm -

numberOfSwaps = 0 // 1tu

for k = 0 to $((n+1)/2)-1$ do // $((n+1)/2) - 1 + 1 = (n+1)/2$ times

 //left to right

 for i = 0 to $2*n-2$ do: // $((2*n-2)-0)/1 + 1 = 2*n-1$ times

 if (disk[i] == D && disk[i+1] == L) // 4 tu

 swap disk[i] and disk[i+1] // 3tu

 numberOfSwaps++ // 1tu

 // $4+\max(4,0) = 8tu$

 endif

 endfor

 //right to left

 for j = $2*n-1$ to 1 do: // $(1 - (2*n-1))/(-1) + 1 = 2*n-2+1 = 2*n-1$ times

 if (disk[j] == L && disk[j-1] == D) // 4tu

 swap disk[j] and disk[j-1] // 3tu

 numberOfSwaps++ //1tu

 // $4+\max(4,0) = 8tu$

 endif

 endfor

endfor

S.C = $((n+1)/2) * (8(2*n-1)+8(2*n-1)) + 1 tu$

 = $((n+1)/2) * (32n - 16) + 1$

 = $16n^2 + 8n - 7$

We want to prove that: $16n^2 + 8n - 7 \in O(n^2)$

So, according to limits theorem,

$\lim_{n \rightarrow \infty} (16n^2 + 8n - 7)/(n^2) \rightarrow (32n+8)/(2n) \rightarrow (32)/(2) = 16.$

Since $16 \geq 0$, and 16 is a constant, the limits theorem tell us that

$16n^2 + 8n - 7 \in O(n^2).$

Alternate Algorithm:

numberOfSwaps = 0 // 1tu

```
for k = 0 to n+1-1 do: // (n/1) + 1 times  
  for i = 0 to 2*n-1 step 2 // (2*n-1)/2 + 1 times  
    if (disk[i] == 'D' && disk[i+1] == 'L') // 4tu  
      swap disk[i] and disk[i+1] // 3tu  
      numberOfSwaps++ // 1tu  
      // 4+max(4,0) = 8tu  
    endif  
  endfor  
  
  // 1 3 5 ... n-2  
  for j = 1 to 2*n-2 step 2 // ((2n-3)/2 + 1) * 8 times  
    if (disk[j] == 'D' && disk[j+1] == 'L') //4 tu  
      swap disk[j] and disk[j+1] // 3tu  
      numberOfSwaps++ // 1tu  
      // 4+max(4,0) = 8tu  
    endif  
  endfor  
Endfor
```

$$\begin{aligned} Sc &= (n/1 + 1) * (((2n-1)/2 + 1)*8 + ((2n-3)/2 + 1) * 8) + 1 \\ &= (n+1) * ((n + \frac{1}{2}) * 8 + (n - \frac{1}{2}) * 8) + 1 \\ &= (n+1) * (8*n + 4 + 8*n - 4) \\ &= (n+1)(16n) + 1 \\ &= 16n^2 + 16n + 1 \text{ tu} \end{aligned}$$

We want to prove that: $16n^2 + 16n + 1 \in O(n^2)$

So, according to limits theorem,

$$\lim_{n \rightarrow \infty} (16n^2 + 16n + 1)' / (n^2)' \rightarrow (32n + 16)' / (2n)' \rightarrow (32)/(2) = 16.$$

Since $16 \geq 0$, and 16 is a constant, the limits theorem tell us that

$$16n^2 + 16n + 1 \in O(n^2).$$