Gerardo Robledo - gerardo.robledo99@csu.fullerton.edu

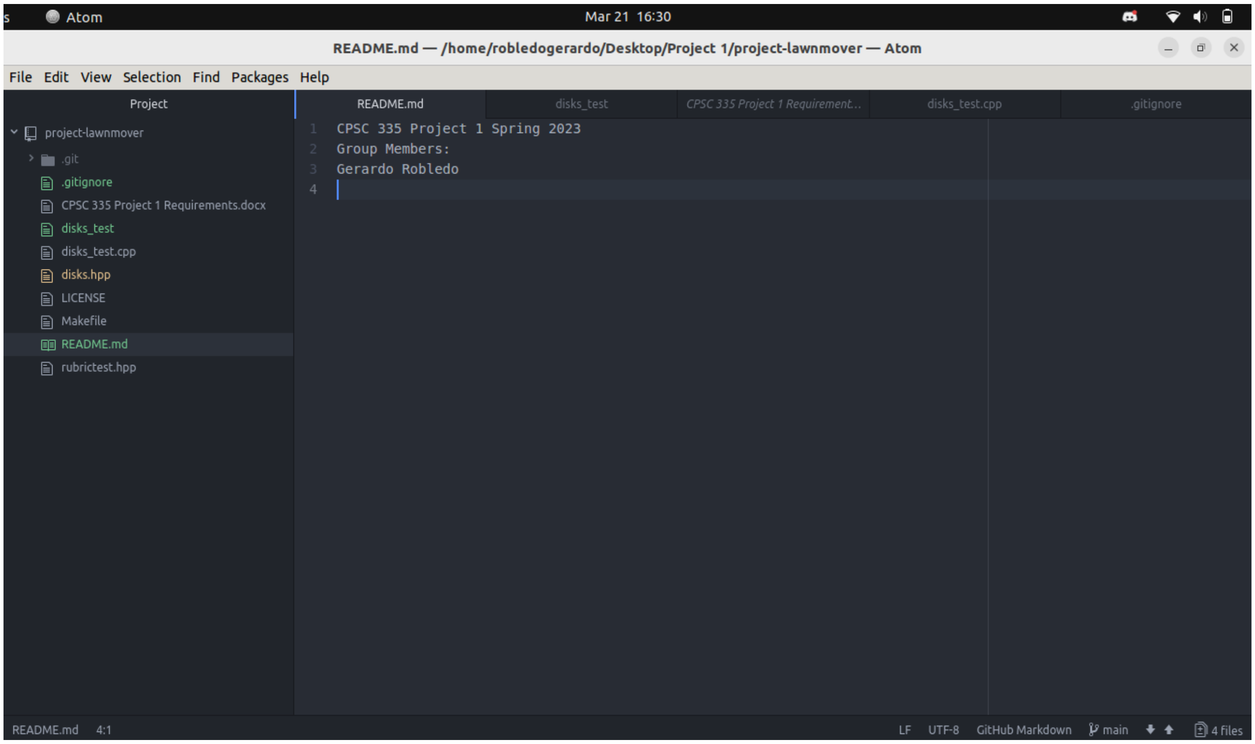
Himani Tawade

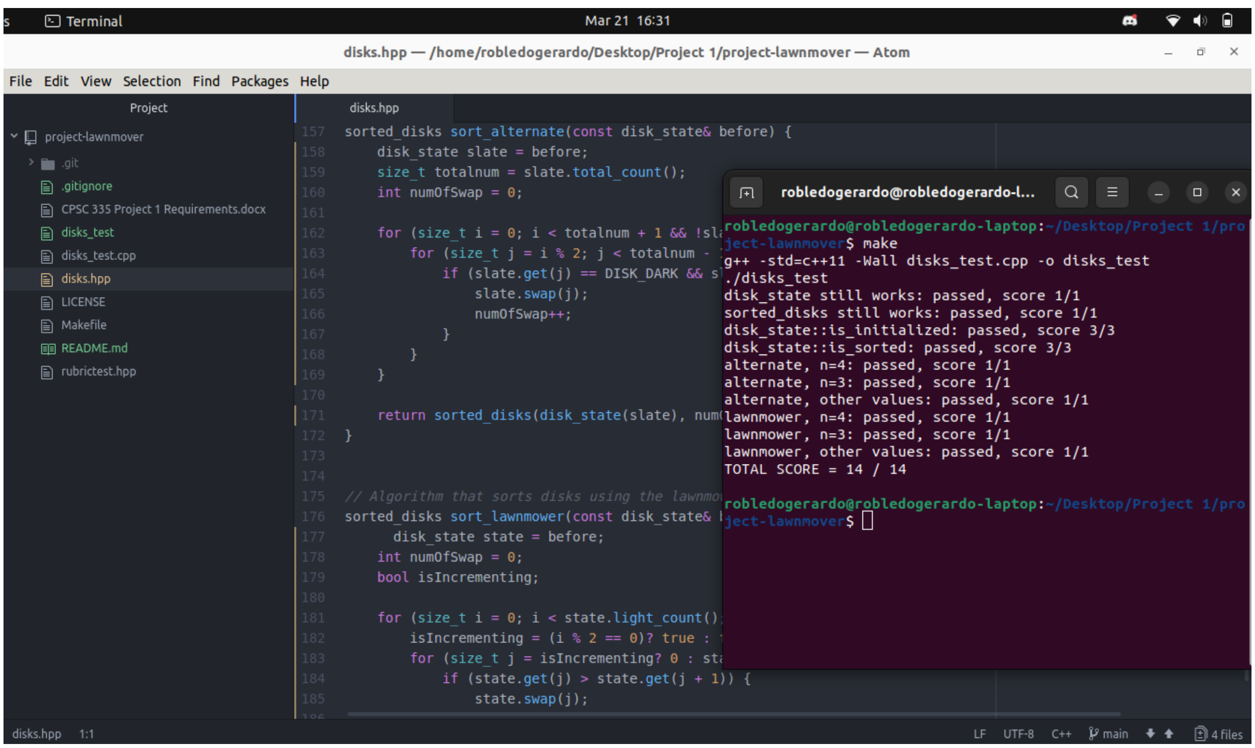
CPSC 335.08

21 March 2023

Project 1

1. For this project, we had to work from code that the professor provided us. We had to create pseudocode from 2 sorting algorithms. Once completing the pseudocode, we had to prove its efficiency class using definition or limi theorem. For our algorithms, we got O(n^2) for both step count. We proved them to be correct using the limit theorem which correctly proved that the time complexity was indeed O(n^2). Once completed, we has to implement our algorithms to the disks.hpp file. We also had to complete the is_sorted function which returns a bool on whether the imputed disks were sorted correctly or not. Once we implemented our psuedocode into the disks.hpp, we compiled and ran the code using the "make" command. Here, it ran our code through the disks_test.cpp file, and gave us our results. Initially we did not get 14/14 but after further investigation of our code, we figured out our errors and got 14/14. We enjoyed working on this project since it gave us real world practice on the material covered in class and also being able to working with partners.

2. 

3. 

4. Pseudocode, step count and proving with limits for sort_algorithm:

Function sort_alternate (disk) {

Step count

Total = length of disk O(1)

numOfswap = 0 (1)

n+1

for (i=0; i<total+1; i++) { O(n)

for (j = i%2; j<total-1; j+=2){ //Determines even or odd O(n)

if (disk(j) Dark and disk(j+1) is light) {

swap disks O(1)

numOfswap++ O(1)

}}}

return disks, numOfswaps

}

$$\in O(n^2)$$

$$\lim_{n\to\infty} \frac{t(n)}{g(n)} \quad \lim_{n\to\infty} \frac{kn^2}{n^2} = \lim_{n\to\infty} k$$

proves since k is going to ∞ as well as n^2 so Time complexity is $\boxed{O(n^2)}$

5. Psuedocode, step count and proving with limits for lawnmover algorithm:

Sort_lawnmower(disks) {

State = disks $O(1)$

bool = isincrementing; $O(1)$

for(i=0; i to light_count(); i++) { $O(n)$

if (isIncreme = i even)

for(j = isincrementing ? 0 : state.total count - 2; $O(n)$

j < total count - 1 ; isincrementing ++j ; --j) {

if(state(j) > state(j+1)) $O(1)$

swap ;

++ numofswap; $O(1)$ $O(1)$

}}

}

return(state num of swaps)

}

$\in O(n^2)$

$\lim_{n \to \infty} \frac{t(n)}{g(n)} = \lim_{n \to \infty} \frac{k \cdot n^2}{n^2} = \lim_{n \to \infty} k$

proves since k is going to ∞ seen as n^2 is so

Time complexity is $\boxed{O(n^2)}$