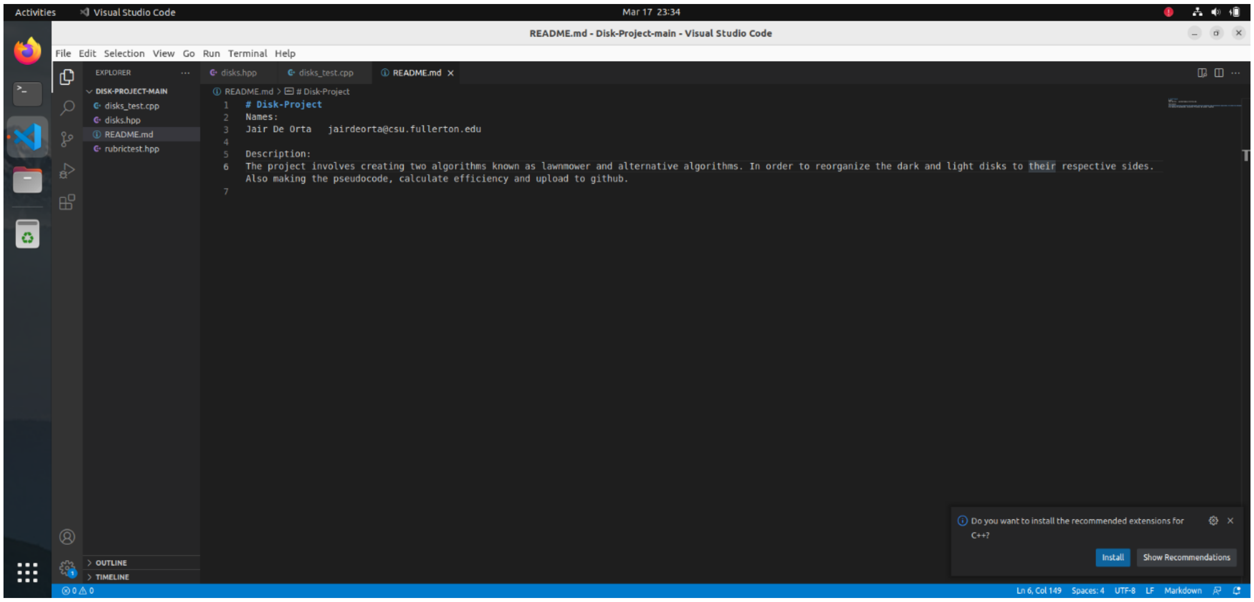
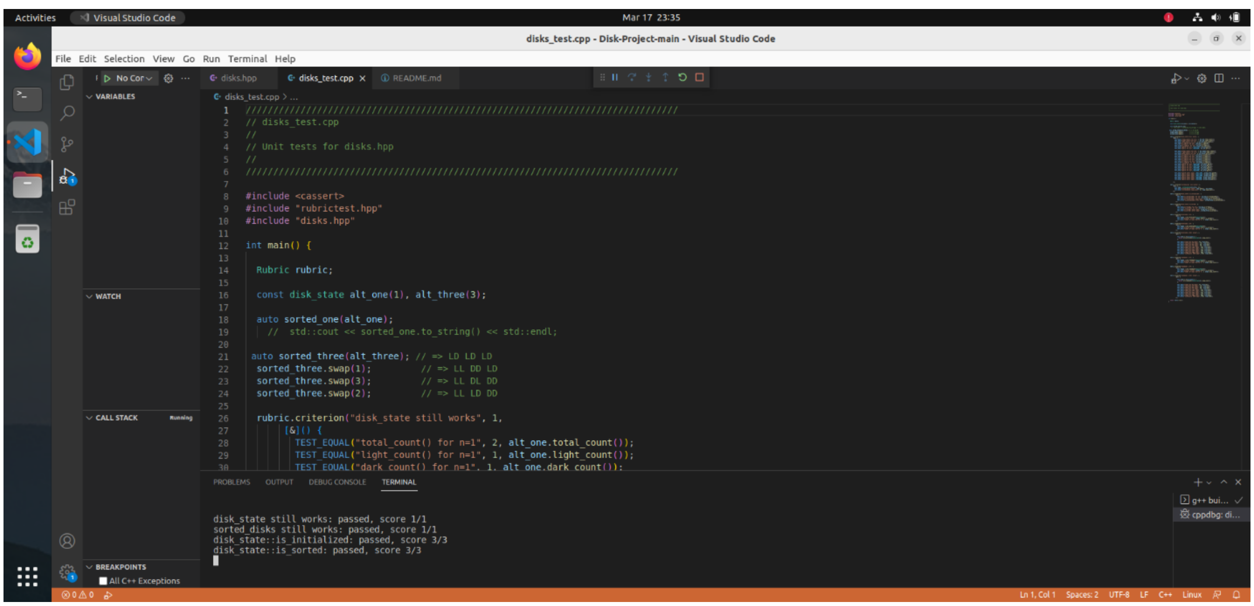
Names: Jair De Orta   jairdeorta@csu.fullerton.edu

This PDF is the submission for Project 1 Lawnmower.

Screenshot of README.md



Screenshot of code running



Pseudocodes

```
def lawnmower(L):
 count is 0 → 1 sc
   for k between 1...n/4 do →n/4 – 1+1=n/4
        for i between 0..2n-2 do →(2n-2-0)+1=2n-1
             if left side == dark and right side == light side → 3sc
                  swap L[i] and L[i+1] and add 1 to count → 2sc
        for i between 2n-2...0 do →(0-(2n-2)/-1)+1=2n-1
             if left side == dark and right side == light side → 3sc
                  swap L[i] and L[i+1] and add 1 to count → 2sc
 return L and count →1sc
```

SC: 1+1+([(2n-1)*5]+ [(2n-1)*5])*(n/4)

=1+1([10n-5]+[10n-5])*(n/4)

=2+(20n-10)*(n/4)

=5n^2-(5/2)n+2 SC

```
def alternative(L):
 count is 0 → 1 sc
    for k between 1...n do → n-1+1=n
        for i between 0..2n-2 in increments of 2 do → ((2n-2)/2)+1= n-1+1=n
             if left side == dark and right side == light side → 3sc
                  swap L[i] and L[i+1] and add 1 to count → 2sc
        for i between 1...2n-2 in increments of 2 do →((2n-2-1)/2)+1=n-3/2+1=n-(1/2)
             if left side == dark and right side == light side → 3sc
                  swap L[i] and L[i+1] and add 1 to count → 2sc
```

return L and count → 1sc

SC=1+1+([(n-1/2)*5]+[n*5])*n

=2+([5n-5/2]+5n)*n

=2+(10n-5/2)*n

=10n^2-(5/2)n+2 SC

Argument for time complexity

Both algorithms have an O(n^2) time complexity. First, an example of O(n^2) is a two nested for loops, and both algorithms are exactly that. But also, using the limit theorem, for lawnmower it becomes 5-0+0=5. It is not infinity, which means it belongs to O(n^2). The same thing applies to the alternative method, being 10-0+0=10. This proof proves that they both have an O(n^2) efficiency class.