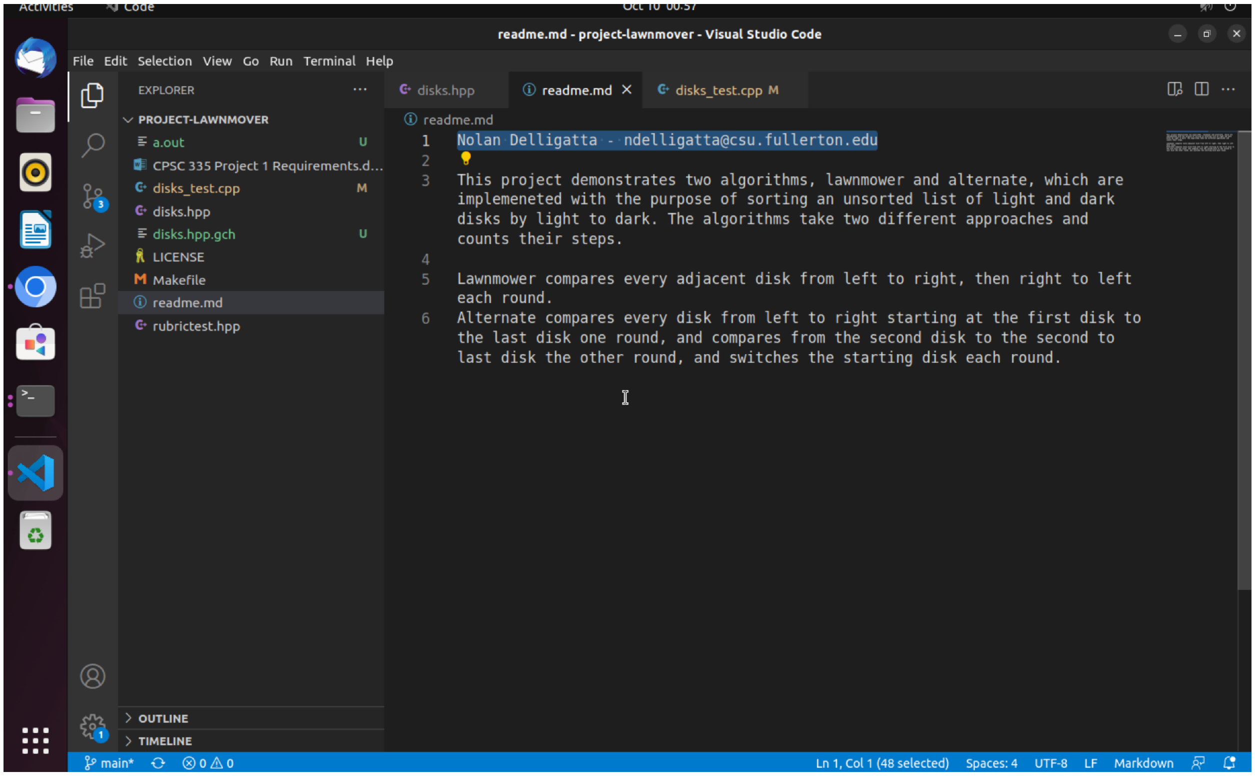
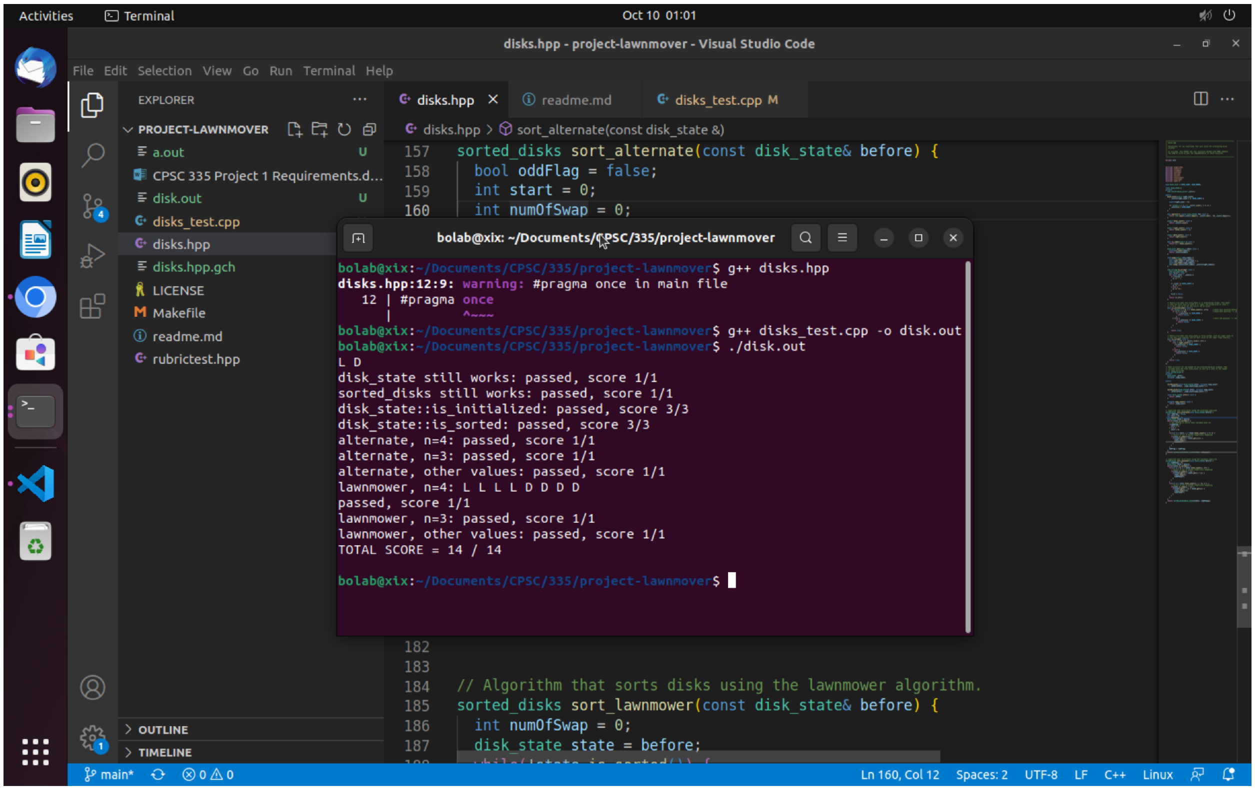
Project 1 PDF

1) Nolan Delligatta - ndelligatta@csu.fullerton.edu

2) Name shown in editor:



3)



4)

Alternate algorithm pseudocode:

Input: A list L of 2*n elements consisting of n light disks and n dark discs

Output: A list S containing the elements of L sorted in order from light disks to dark discs

```
def alternate(L):
 swaps = 0 <- 1 tu
 start = 0 <- 1 tu
 oddFlag = false <- 1 tu
 S = L <- 1 tu
 while(S is not sorted): <- (n + 1) tu
   if(oddFlag): <- 1 + max(1, 1) tu
     start = 1 <- 1 tu
   else:
     start = 0 <- 1 tu
   for(i = start; i < len(S); i += 2): <- n / 2 tu
     if(i+1 < len(S)): <- 2 + max(2 + max(2, 0), 0) tu
       if(S[i] > S[i + 1]): <- 2 + max(3, 0) tu
         S.swap(i, i + 1) <- 2 tu
         swaps++ <- 1 tu
   oddFlag = !oddFlag <- 1 tu
```

return S

Step Count = 4 + SC(While)
SC(While) = (n + 1) * SC(If/Else)
SC(If/Else) = 1 + max(1,1) = 1 + 1 = 2

SC(While) = (n + 1) * (2 + SC(For))
SC(For) = (n / 2) * SC(If) = (n / 2) * 7
SC(If) = 2 + max(2 + max(3, 0), 0) = 2 + max(2 + 3, 0) = 2 + max(5, 0) = 2 + 5 = 7
SC(For) = (n / 2) * 7

SC(While) = (n + 1) *(2 + ((n / 2) * 7) + 1) = (n + 1) *(((n / 2) * 7) + 3)

Step Count = 4 + (n + 1) *((($\frac{n}{2}$) * 7) + 3) = $\frac{7}{2}n^2 + \frac{13}{2}n + 7$

Lawnmower algorithm pseudocode:

Input: A list L of 2*n elements consisting of n light disks and n dark discs
Output: A list T containing the elements of L sorted in order from light disks to dark discs

```
def lawnmower(L):
  swaps = 0 <- 1 tu
  T = L <- 1 tu
  while(T is not sorted): <- n / 2 tu
    for(i = 0; i < len(T); i++): <- n tu
      if(i + 1 < len(T)): <- 2 + max(2 + max(3,0),0))
        if(T[i] > T[i + 1]): <- 2 + max(3, 0)
          T.swap(i, i+1) <- 2 tu
          steps++ <- 1 tu
    for(j = len(T); i > 0; i--): <- n tu
        if(j - 1 > 0): <- 2 + max(2 + max(3,0),0))
          if(T[j - 1] > T[j]): <- 2 + max(3,0)
            T.swap(j - 1, j) <- 2 tu
            steps++ <- 1 tu
  return T
```

Step count = 2 + SC(While)
SC(While) = (n / 2) * SC(For)
SC(For) = n * SC(If) = n * 7
SC(If) = 2 + max(2 + max(3,0),0)) = 2 + max(2 + 3, 0) = 2 + 5 = 7

SC(While) = (n / 2) * SC(For) = (n / 2) * (7n + SC(For))

SC(For) = n * SC(If) = n * 7
SC(If) = 2 + max(2 + max(3,0),0)) = 2 + max(2 + 3, 0) = 2 + 5 = 7
SC(While) = (n / 2) * SC(For) = (n / 2) * (7n + 7n) = (n / 2) * (14n)
Step count = 2 + SC(While) = 2 + (n / 2) * (14n) = $7n^2 + 2$

5)
Alternate algorithm proof:
$\frac{7}{2}n^2 + \frac{13}{2}n + 7 \in O(n^2)$

$lim_{n->\infty} \frac{\frac{7}{2}n^2+\frac{13}{2}n+7}{n^2}$
LH= $lim_{n->\infty} \frac{7n + \frac{13}{2}}{2n}$
LH=$lim_{n->\infty} \frac{7}{2} = \frac{7}{2} \geq 0$
By limit theorem, $\frac{7}{2}n^2 + \frac{13}{2}n + 7 \in O(n^2)$

Lawnmower algorithm proof:
$7n^2 + 2 \in O(n^2)$

$lim_{n->\infty} \frac{7n^2+2}{n^2}$
LH= $lim_{n->\infty} \frac{14n}{2n}$
LH= $lim_{n->\infty} \frac{14}{2} = 7 \geq 0$
By limit theorem, $7n^2 + 2 \in O(n^2)$