Alternate and Lawnmower Project 1 PDF Report

Submission:

This report and project submission is for project 1 for Computer Science Algorithms 335.

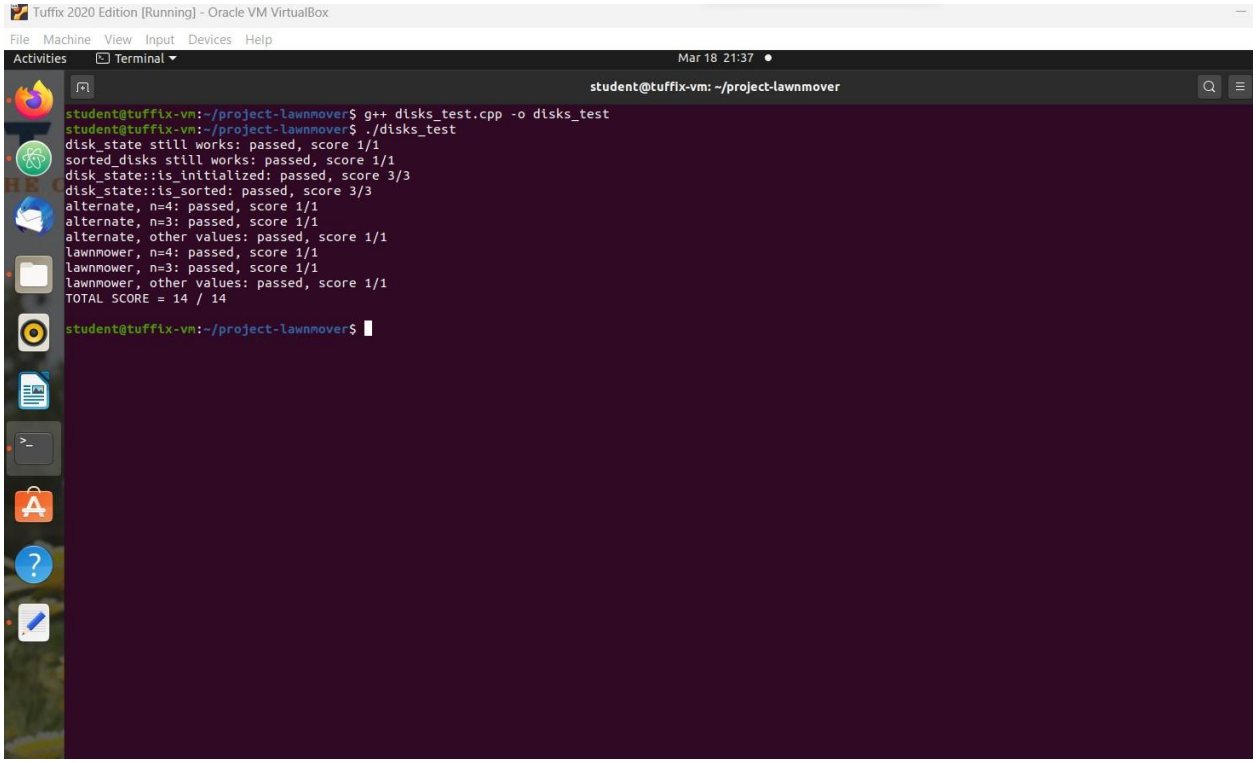
Contributors:

Ricardo Granados Macias: ricardog2002@csu.fullerton.edu

Tommy Ly: lytommy321@csu.fullerton.edu

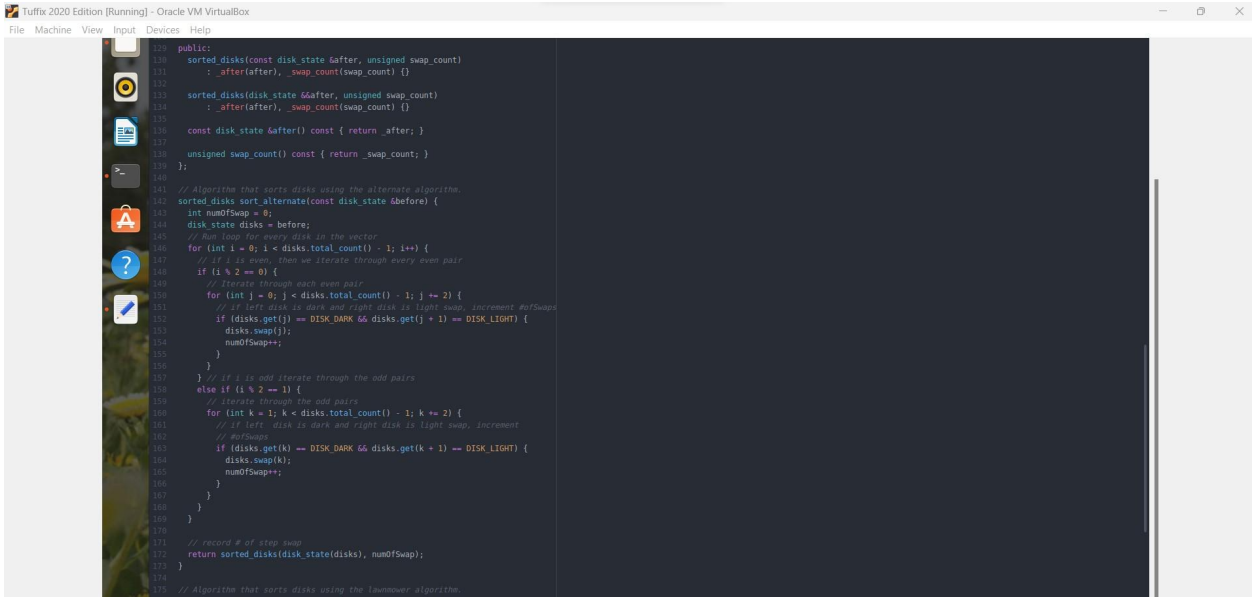
Project Screenshots:

Program Compiling & Passing Tests:

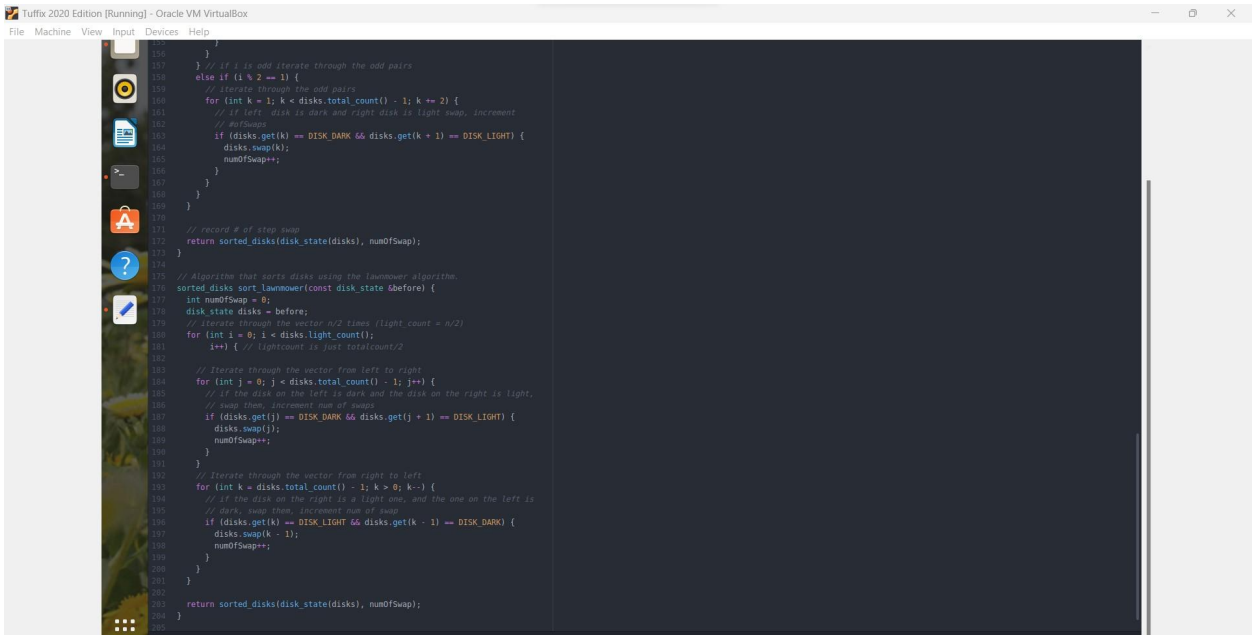


Is Sorted Function:

Alternate Sort Algorithm:

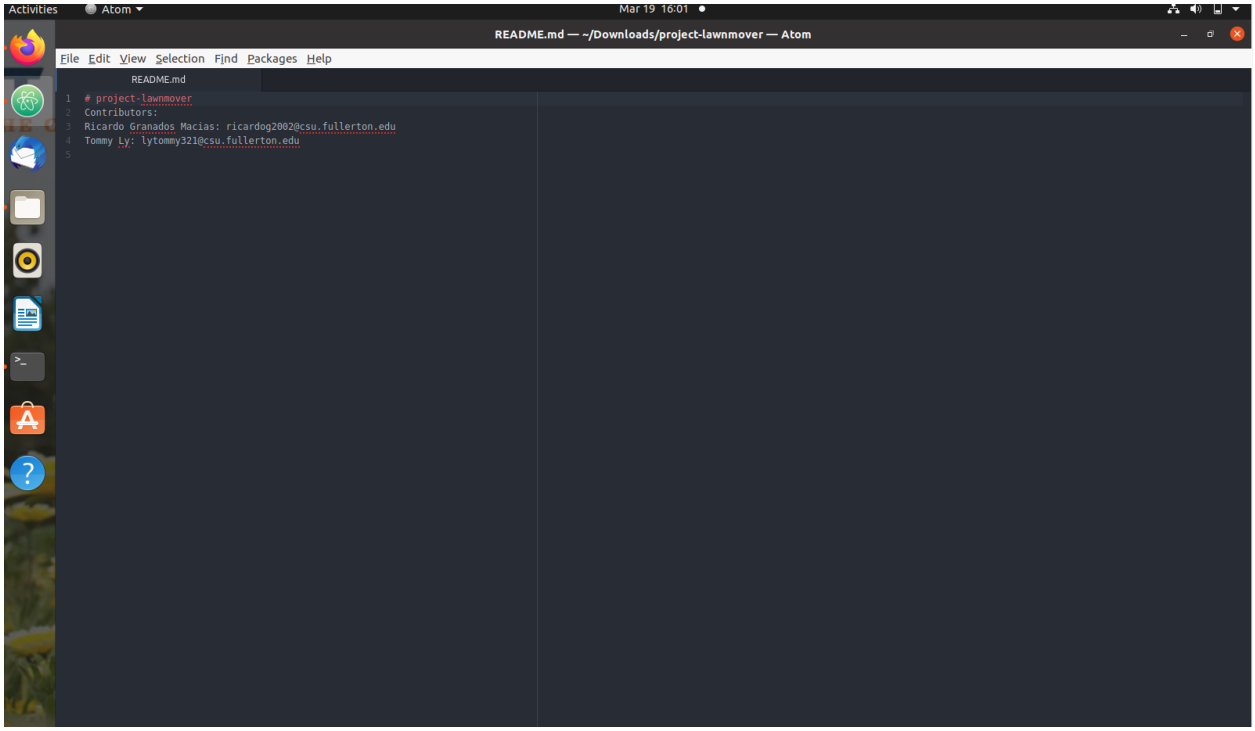


Lawnmower Algorithm:




```
    } // if i is odd iterate through the odd pairs
    else if (i % 2 == 1) {
      // iterate through the odd pairs
      for (int k = 1; k < disks.total_count() - 1; k += 2) {
        // if left disk is dark and right disk is light swap, increment
        // #ofSwaps
        if (disks.get(k) == DISK_DARK && disks.get(k + 1) == DISK_LIGHT) {
          disks.swap(k);
          numOfSwap++;
        }
      }
    }
  }

  // record # of step swap
  return sorted_disks(disk_state(disks), numOfSwap);
}

// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state &before) {
  int numOfSwap = 0;
  disk_state disks = before;
  // iterate through the vector n/2 times (light_count = n/2)
  for (int i = 0; i < disks.light_count();
       i++) { // lightcount is just totalcount/2

    // Iterate through the vector from left to right
    for (int j = 0; j < disks.total_count() - 1; j++) {
      // if the disk on the left is dark and the disk on the right is light,
      // swap them, increment num of swaps
      if (disks.get(j) == DISK_DARK && disks.get(j + 1) == DISK_LIGHT) {
        disks.swap(j);
        numOfSwap++;
      }
    }
    // Iterate through the vector from right to left
    for (int k = disks.total_count() - 1; k > 0; k--) {
      // if the disk on the right is a light one, and the one on the left is
      // dark, swap them, increment num of swap
      if (disks.get(k) == DISK_LIGHT && disks.get(k - 1) == DISK_DARK) {
        disks.swap(k - 1);
        numOfSwap++;
      }
    }
  }

  return sorted_disks(disk_state(disks), numOfSwap);
}
```

ReadME FILE:




```
# project-lawnmover
Contributors:
Ricardo Granados Macias: ricardog2002@csu.fullerton.edu
Tommy Ly: lytommy321@csu.fullerton.edu
```

Pseudocode:

Alternate Algorithm Pseudocode:

```
I/P: Unsorted vector of disks, can't be empty, size n
O/P: Sorted vector of disks, can’t be empty, size n
/* def alternate_sort(unsorted) :
sorted = unsorted     // 1TU
numSwap = 0          // 1TU
for i = 0 to n - 1 do: // (n - 1 + 1) TU
     if ( i % 2 == 0) then  // ( 2 TU)
         for j = 0 to n - 1 step 2 do:   // ( (n - 1) /2 ) + 1 TU
             if (left == dark && right == light) then  // 3 TU
                 swap(left, right)   // 1 TU
                 numSwap++    // 1 TU
             endif
          end innerfor
       end if
    if ( i % 2 == 1) do:   // 2 TU
        for k = 1 to n - 1 step 2 do:   // ((n -1 -1) / 2) + 1 TU
            if (left == dark && right == light) then // 3 TU
                swap(left, right)    // 1 TU
                numSwap++     // 1 TU
            endif
         end innerfor
        end if
  end outerfor
return (sorted, numSwap)  // 0 TU


*/
```

Alternate Algorithm Step Count Calculaition:

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

SC = 2 + n(5n + 13/2)
SC = 2 + (5n^2) + 13n/2

Lawnmower Algorithm Pseudocode:

```
This is the lawnmower pseudoCode in a c++ coding format
//I/P unsorted list of colored disks that is not empty, size n
// O/P sorted list of colored disks, size n, not empty
/*
def lawn_mover_sort(unsorted) :
numSwaps = 0; // 1 TU
sorted_disks = unsorted // 1 TU
for i = 0 to n/2 do:    // (n/2 - 0) + 1 TU
    for i = 0  to n -1 do: //  (n - 1 - 0) + 1 TU
        if (left == dark && right == light) then: // 3TU
             swap(left,right)     // 1 TU
            sorted_disks.numSwaps++; // 1 TU
        end if
    end innerfor1
     for i = n -1 to 0 do:   // ((0 - (n -1)) / -1 ) + 1
         if (right == light && left == dark) then: // 3TU
             sorted_disks.swap(left,  right) // 1 TU
             numSwaps++;   //  1 TU
        Endif
      end innerfor2
 End outerfor
return (sorted_disks, numSwaps)  // 0  TU

*/
```

Lawnmover Algorithm Step Count Calculaition:

Step Count = 2 + ((n/2 + 1) * (5n + 5n))
Step Count = 2 +((n/2 + 1) * (10n))
Step Count = 10n^2 /2 + 10n + 2
Step Count = 5n^2 + 10n + 2

Time Complexity Proofs:

Alternate Algorithm Analysis / Proof:

Assume f(n) = $5n^2 + 13n/2 + 2$

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

We will prove that f(n) is within g(n) with a proof using limits

Proof by limit:

Limit ($(5n^2 + 13n/2 + 2) / n^2$
$n \rightarrow \infty$

Divide by the greatest common denominator

Limit ($5n^2/ n^2 + 13n /2 n^2 + 2 / n^2$) / n^2 / n^2
$n \rightarrow \infty$

Simplify to get

Limit ($5 + 13/2n + 2/n^2$)
$n \rightarrow \infty$

Substitute infinity for n to get

Limit $5 + 13/\infty + 2/\infty$
$n \rightarrow \infty$

Simplify

Limit $5 + 0 + 0 = 5$
$n \rightarrow \infty$

Therefore, because we get a real number 5, we can conclude by the limit theorem that f(n) is within $O(n^2)$. Therefore, our algorithm is within $O(n^2)$.

Lawnmower Algorithm Analysis /Proof:

Assume f(n) = 5n^2 + 10n + 2
Assume g(n) = O(n^2)
We will prove that f(n) is within g(n) with a proof using limits

Proof by limit:

Limit (5n^2 + 10n + 2) / n^2
$n \to \infty$

Divide by the greatest common denominator

Limit (5n^2/ n^2+ 10n / n^2 + 2 / n^2) / n^2 / n^2
$n \to \infty$

Simplify to get

Limit (5 + 10/n + 2/n^2)
$n \to \infty$

Substitute infinity for n to get

Limit 5 + 10/∞ + 2/∞
$n \to \infty$

Simplify

Limit 5 + 0 + 0 = 5
$n \to \infty$

Therefore, because we get a real number 5, we can conclude by the limit theorem that f(n) is within O(n^2). Therefore, our lawnmover algorithm is within O(n^2).