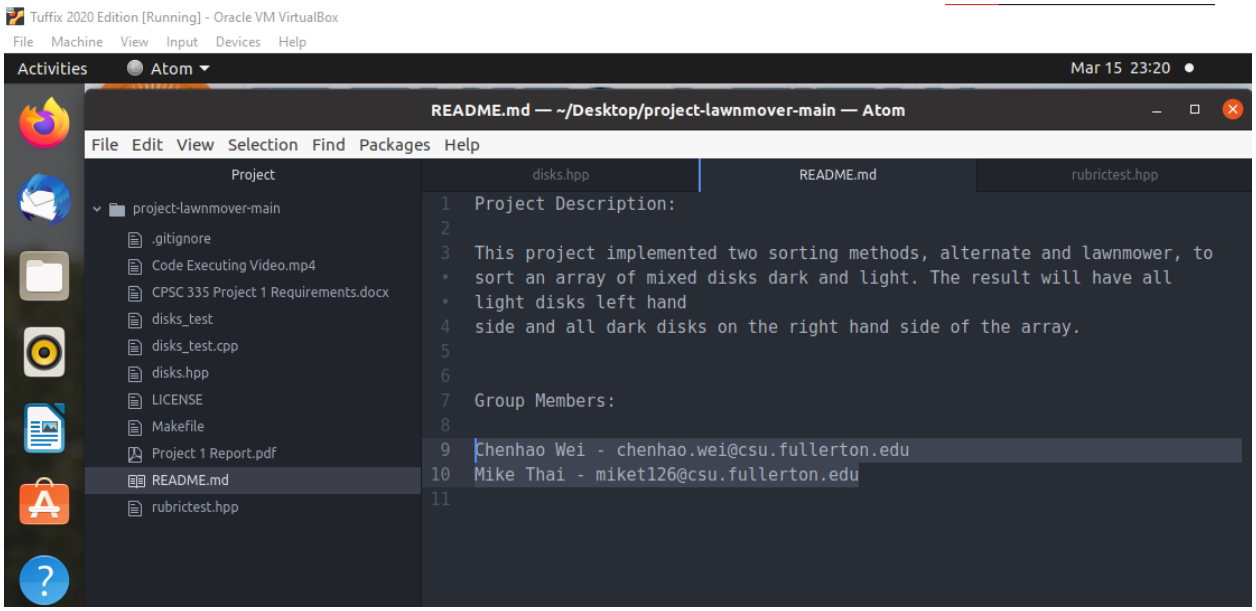
Project 1

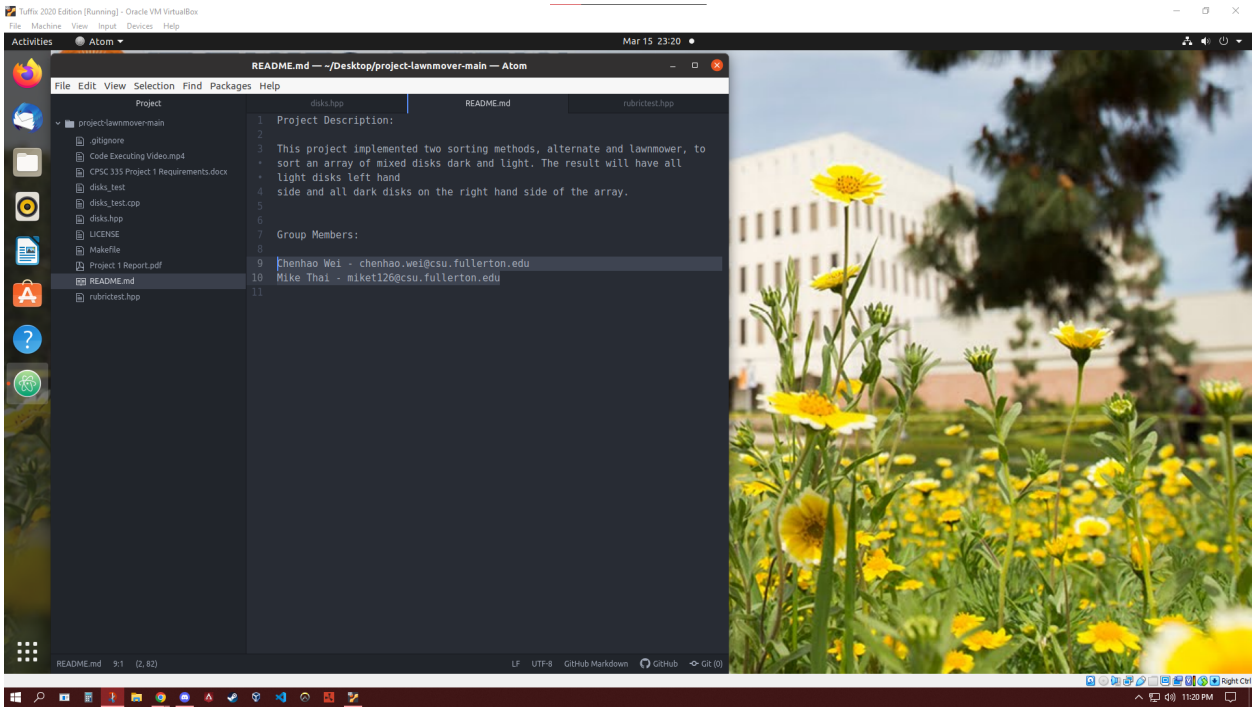
Mike Thai - miket126@csu.fullerton.edu

Chenhao Wei - chenhao.wei@csu.fullerton.edu

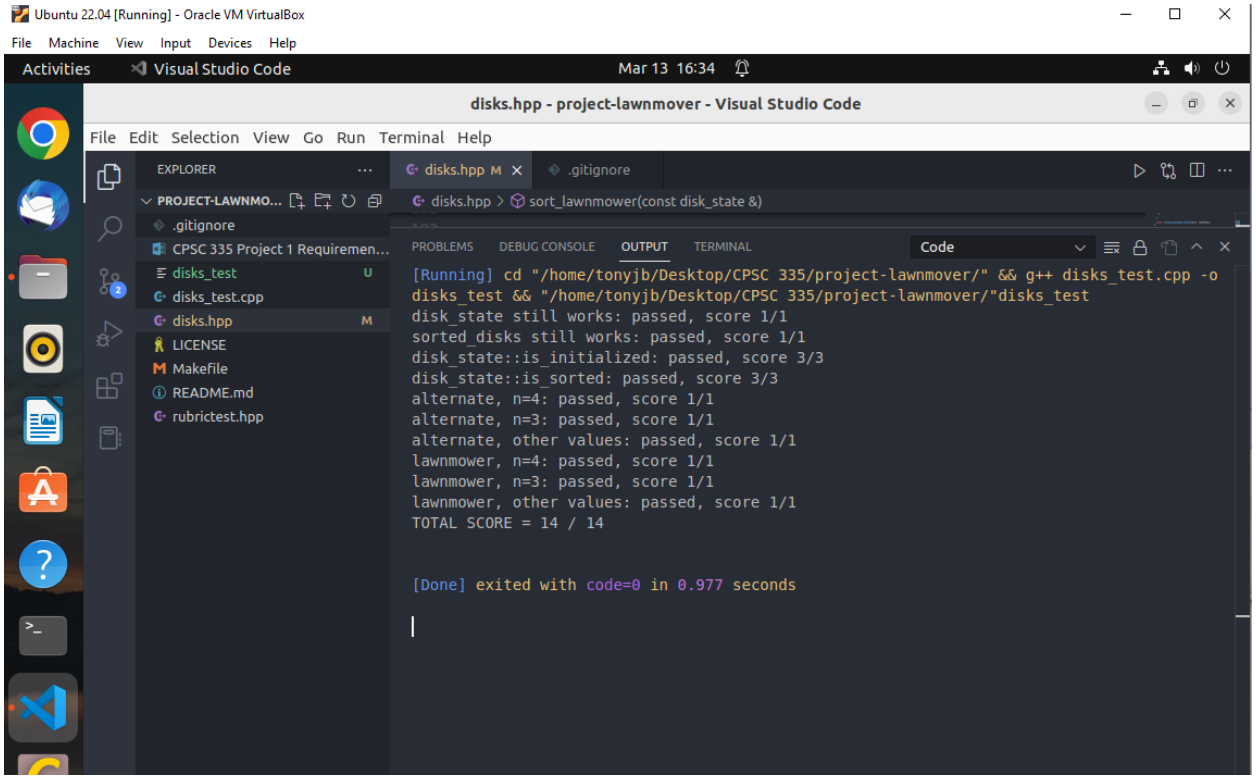
Due: March 19, 2023

README Screenshot

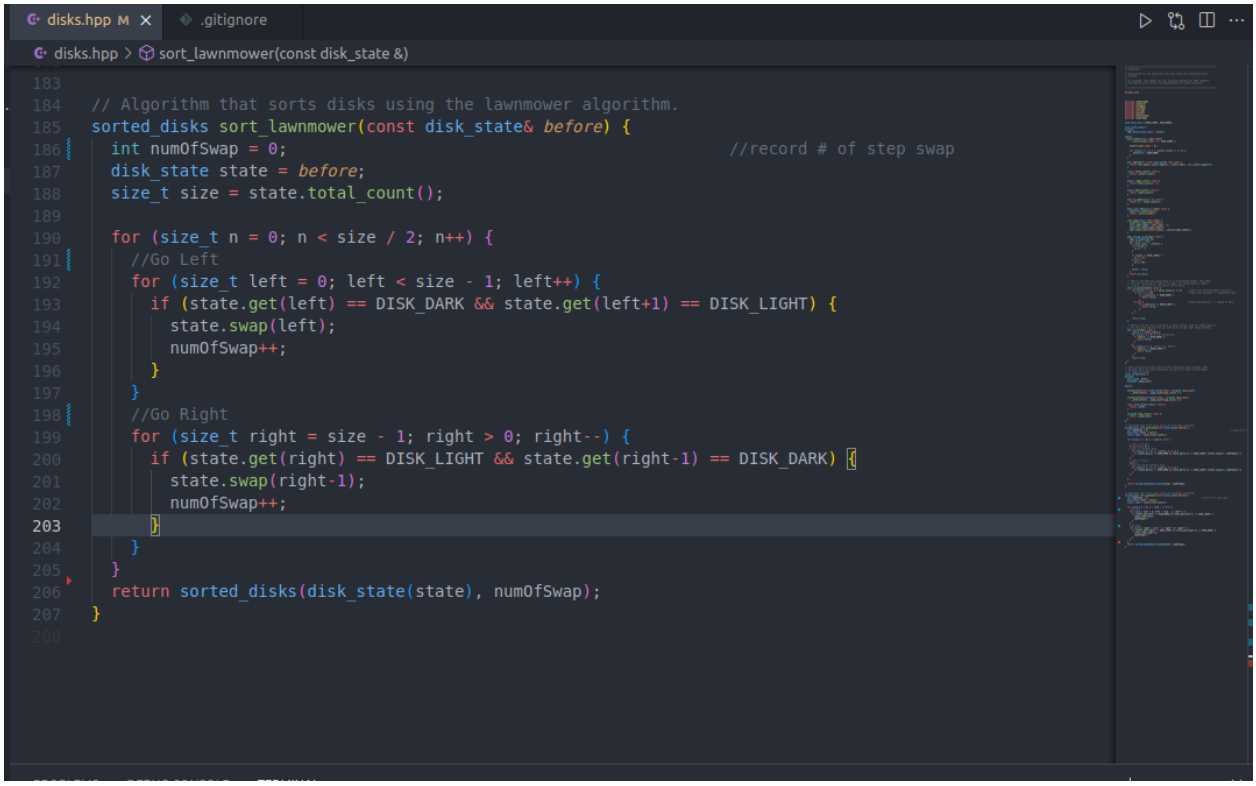




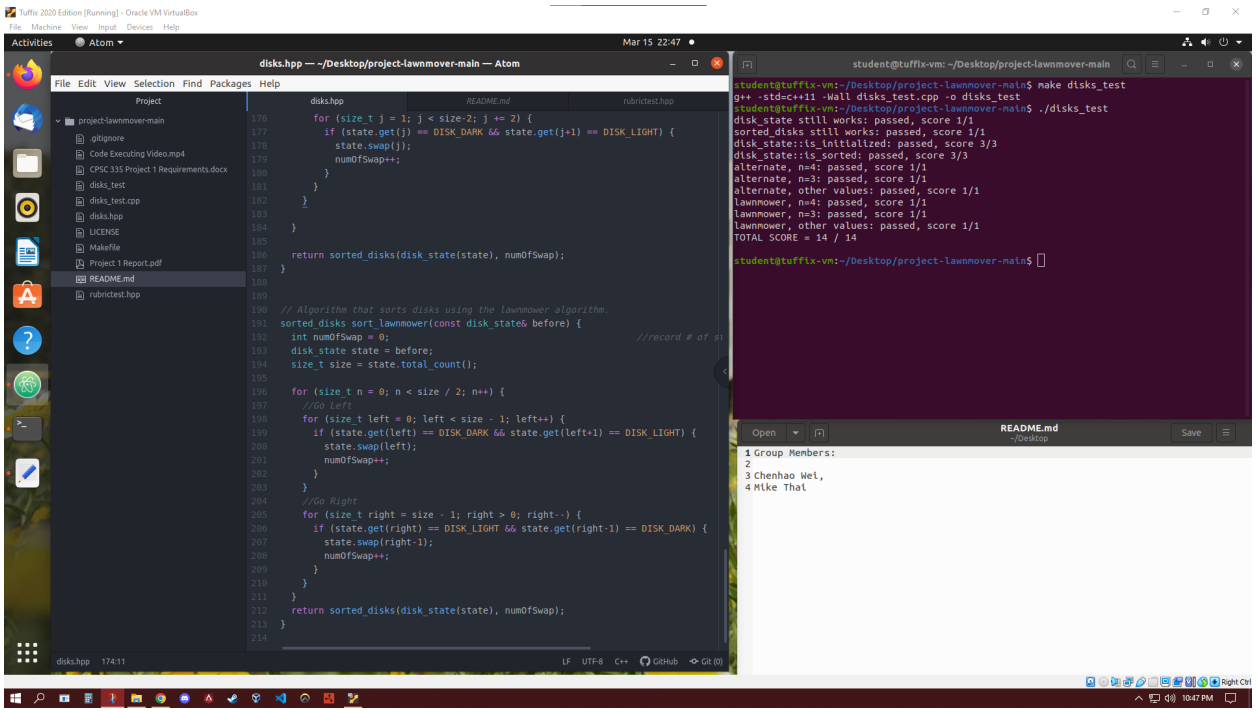
Code Compiling and Executing Screenshot



Lawnmower Algorithm



```cpp
// Algorithm that sorts disks using the lawnmower algorithm.
sorted_disks sort_lawnmower(const disk_state& before) {
  int numOfSwap = 0;                                    //record # of step swap
  disk_state state = before;
  size_t size = state.total_count();

  for (size_t n = 0; n < size / 2; n++) {
    //Go Left
    for (size_t left = 0; left < size - 1; left++) {
      if (state.get(left) == DISK_DARK && state.get(left+1) == DISK_LIGHT) {
        state.swap(left);
        numOfSwap++;
      }
    }
    //Go Right
    for (size_t right = size - 1; right > 0; right--) {
      if (state.get(right) == DISK_LIGHT && state.get(right-1) == DISK_DARK) {
        state.swap(right-1);
        numOfSwap++;
      }
    }
  }
  return sorted_disks(disk_state(state), numOfSwap);
}
```



```
student@tuffix-vm:~/Desktop/project-lawnmover-main$ make disks_test
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
student@tuffix-vm:~/Desktop/project-lawnmover-main$ ./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

student@tuffix-vm:~/Desktop/project-lawnmover-main$
```

```
Group Members:
2
3 Chenhao Wei,
4 Mike Thai
```

Lawnmover

```
def sort_lawnmover(disk_state):
    number of swap = 0                                  +1
        n = disk_state.size()                           +1
    for i=1    to  n/2 :
        for left=1 to n-1 :
            if (left color == Dark && right color == light)   +3
                swap
                number of swap++;                               +1
            end if
        end for

        for right=n-1  to  1 :
            if (left color == Dark && right color == light)   +3
                swap
                number of swap++;                               +1
            end if
        end for

    end for
return;
```

Step count:

$$2+\sum_{i=1}^{\frac{n}{2}}\left(\sum_{1}^{n-1}4+\sum_{1}^{n-1}4\right)$$

$$=2+\sum_{i=1}^{\frac{n}{2}}\left(4(n-1-1+1)+4(n-1-1+1)\right)$$

$$=2+\sum_{i=1}^{\frac{n}{2}}(8n-8)$$

$$=2+(8n-8)\left(\frac{n}{2}-1+1\right)$$

$$=4n^2-4n+2$$

Prove $4n^2-4n+2 \in O(n^2)$

take $\frac{4n^2-4n+2}{n^2} = 4 - \frac{4}{n} + \frac{2}{n^2}$

as $n \to \infty$

$4 \to 4$

$\frac{4}{n} \to 0$

$\frac{2}{n^2} \to 0$

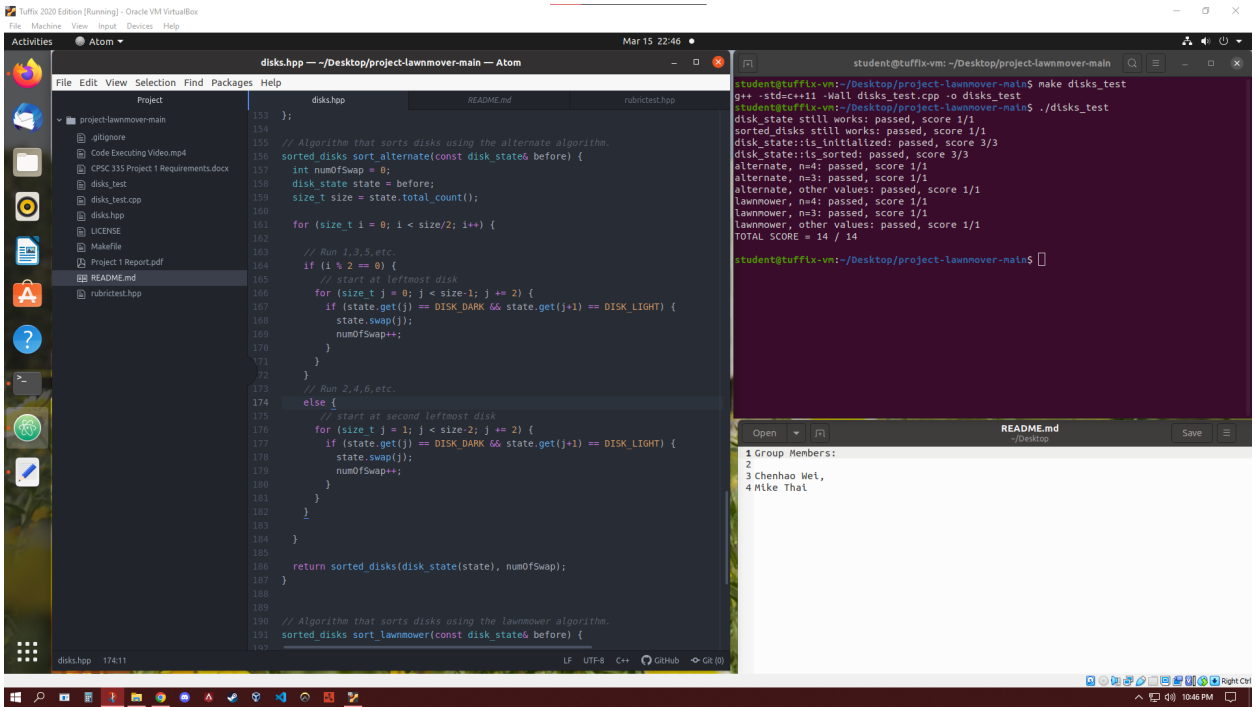
hence $\frac{4n^2-4n+2}{n^2} \to 4 \geq 0$

Therefore $4n^2-4n+2 \in O(n^2)$

Alternate Algorithm

```cpp
// Algorithm that sorts disks using the alternate algorithm.
sorted_disks sort_alternate(const disk_state& before) {
  int numOfSwap = 0;
  disk_state state = before;
  size_t size = state.total_count();

  for (size_t i = 0; i < size/2; i++) {

    if (i % 2 == 0) {
      for (size_t j = 0;j < size; j += 2) {
        if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
          state.swap(j);
          numOfSwap++;
        }
      }
    }
    else {
      for (size_t j = 1; j < size-1; j += 2) {
        if (state.get(j) == DISK_DARK && state.get(j+1) == DISK_LIGHT) {
          //state[j] = DISK_LIGHT;
          //state[j+1] = DISK_DARK;
          state.swap(j);
          numOfSwap++;
        }
      }
    }
  }

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

```
    score 0/3
disks_test: disks.hpp:70: void disk_state::swap(size_t): Assertion `is_index(right_
Aborted (core dumped)
student@tuffix-vm:~/Desktop/project-lawnmover-main$ make disks_test
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
student@tuffix-vm:~/Desktop/project-lawnmover-main$ ./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:
    TEST FAILED:
    line 81 of file disks_test.cpp, message: is_sorted() for n=3
    score 0/3
alternate, n=4: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=4:
    TEST FAILED:
    line 123 of file disks_test.cpp, message: number of swaps must be 6
    score 0/1
lawnmower, n=3:
    TEST FAILED:
    line 130 of file disks_test.cpp, message: number of swaps must be 3
    score 0/1
lawnmower, other values:
    TEST FAILED:
    line 140 of file disks_test.cpp, message: n=10 gives 45 swaps
    score 0/1
TOTAL SCORE = 8 / 14

student@tuffix-vm:~/Desktop/project-lawnmover-main$
```



Alternate Algo

List of disks, L

Size of list, 2n

```
count = 0                                   — 1 tu
for (i = 0 to n)
    if (i % 2 == 0)                         — 2 tu
        for (j = 0 to 2n-1) step 2
            if (L[j] == dark && L[j+1] == light) — 4 tu
                L[j] = light
                L[j+1] = dark               } 4 tu
                count ++
            endif
        endfor
    else
        for (j = 1 to 2n-2) step = 2
            if (L[j] == dark && L[j+1] == light) — 4 tu
                L[j] = light
                L[j+1] = dark               } 4 tu
                count ++
            endif
        endfor
    endif
endfor
return count                                — 0 tu
```

Alternate Algo SC

$SC_{inner\ if\ 1} = 4 + \max(4, 0) = 8$

$SC_{inner\ if\ 2} = 4 + \max(4, 0) = 8$

$SC_{inner\ for\ 1} = iteration * SC_{inner\ if\ 1}$

$= \left(\frac{2n - 1 - 0}{2} + 1\right) * 8 = \left(n + \frac{1}{2}\right) * 8 = 8n + 4$

$SC_{inner\ for\ 2} = iteration * SC_{inner\ if\ 2}$

$= \left(\frac{2n - 2 - 1}{2} + 1\right) * 8 = \left(n - \frac{1}{2}\right) * 8 = 8n - 4$

$SC_{outer\ if} = 2 + \max(SC_{inner\ for\ 1}, SC_{inner\ for\ 2})$

$= 2 + \max(8n + 4, 8n - 4) = 6 + 8n$

$SC_{outer\ for} = iteration * SC_{outer\ if}$

$= (n - 0 + 1) * (6 + 8n)$

$= 6n + 6 + 8n^2 + 8n = 8n^2 + 14n + 6$

$SC_{Alternate} = 1 + SC_{outer\ for}$

$= 1 + 8n^2 + 14n + 6$

$= 8n^2 + 14n + 7$

Alternate Algo Efficiency Class

Prove: $8n^2 + 14n + 7 \in O(n^2)$

By def:

$$8n^2 + 14n + 7 \leq C \cdot n^2 \quad \forall n \geq n_0$$

(Choose: $C = 8 + 14 + 7 = 29$

$n_0 = 1$)

$\rightarrow 8n^2 + 14n + 7 \leq 29n^2$

$8 \cdot 1 + 14 \cdot 1 + 7 \leq 29 \cdot 1$

$29 \leq 29 \rightarrow$ True

$\rightarrow 8n^2 + 14n + 7 \in O(n^2)$

$\Rightarrow$ Alternate Algo Time Efficiency is $O(n^2)$