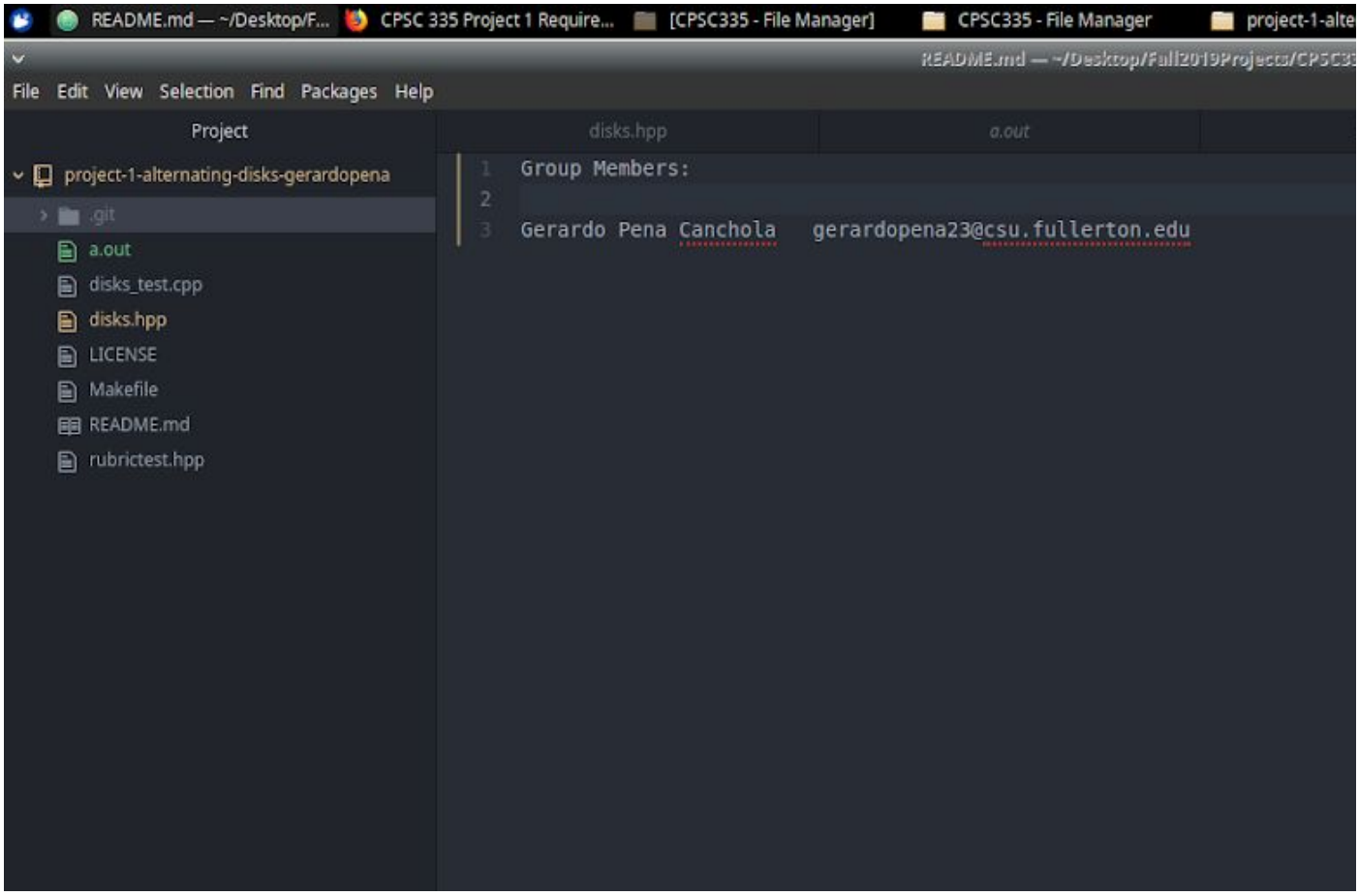
Gerardo Pena Canchola
gerardopena23@csu.fullerton.edu

Project 1



The left-to-right algorithm

```
for (i = 0; i < sizeOfVector - 1; i++) {
        for (j = 0; j < sizeOfVector - 2; j++) {                    // (n - 2 + j + 1) * 4 = n - j - 1
                if (copy.get(j) > copy.get(j + 1)) {                // 1 step * 2
                        copy.swap(j);                               // 1 step
                        count++;                                    // 1 step * 2
                                                                    //2 + max(1*2, 0) = 4
                }
        }
                                                                    // 4n - 4j - 4 = 4n^2 - 4(n(n+1)/4) -
4n
                                                                    //= 0(n^2)
}
```

Gerardo Pena Canchola
gerardopena23@csu.fullerton.edu

Lawnmower algorithm

```
do {
        check = false;                                          // 1 step * 1

        for (int i = 0; i < sizeOfVector - 2; i++) {            // n - 2 - i + 1 = (n - i - 1) * 4
                if (copy.get(i) > copy.get(i + 1)) {            // 1 step * 2
                        copy.swap(i);                           // 1 step * 1
                        count++;                                // 1 step * 2
                        check = true;                           // 1 step * 1
                        // 2 + max(1*2*1) = 4
                }
                                                                // 4(n - i - 1) = 4n - 4i - 4
        }

        if (!check) {    // 1 step * 1
                break;                                          // 1 step * 1
        }

                                                                // 1 + max(1,0) = 2

        check = false;                                          // 1 step * 1

        for (int i = sizeOfVector - 2; i = 0; i--) {            // n - 2 - i = 4(n - i - 1)
                if (copy.get(i) > copy.get(i + 1)) {            // 1 step * 2
                        copy.swap(i);                           // 1 step * 1
                        count++;                                // 1 step * 2
                        check = true;                           // 1 step * 1
                }
                                                                // 2 * max(1*2*1, 0) = 4
        }

} while (check);    // 1 step * 1

                                                                // 2(4n - 4i - 4) + 1 + 1 + 2 =
                                                                // 8n - 8i - 8 + 4 = 8n - 8i - 4 =
                                                                // 8n^2 - 8(n(n+1)/8) - 4n =
                                                                // 0(n^2)
```