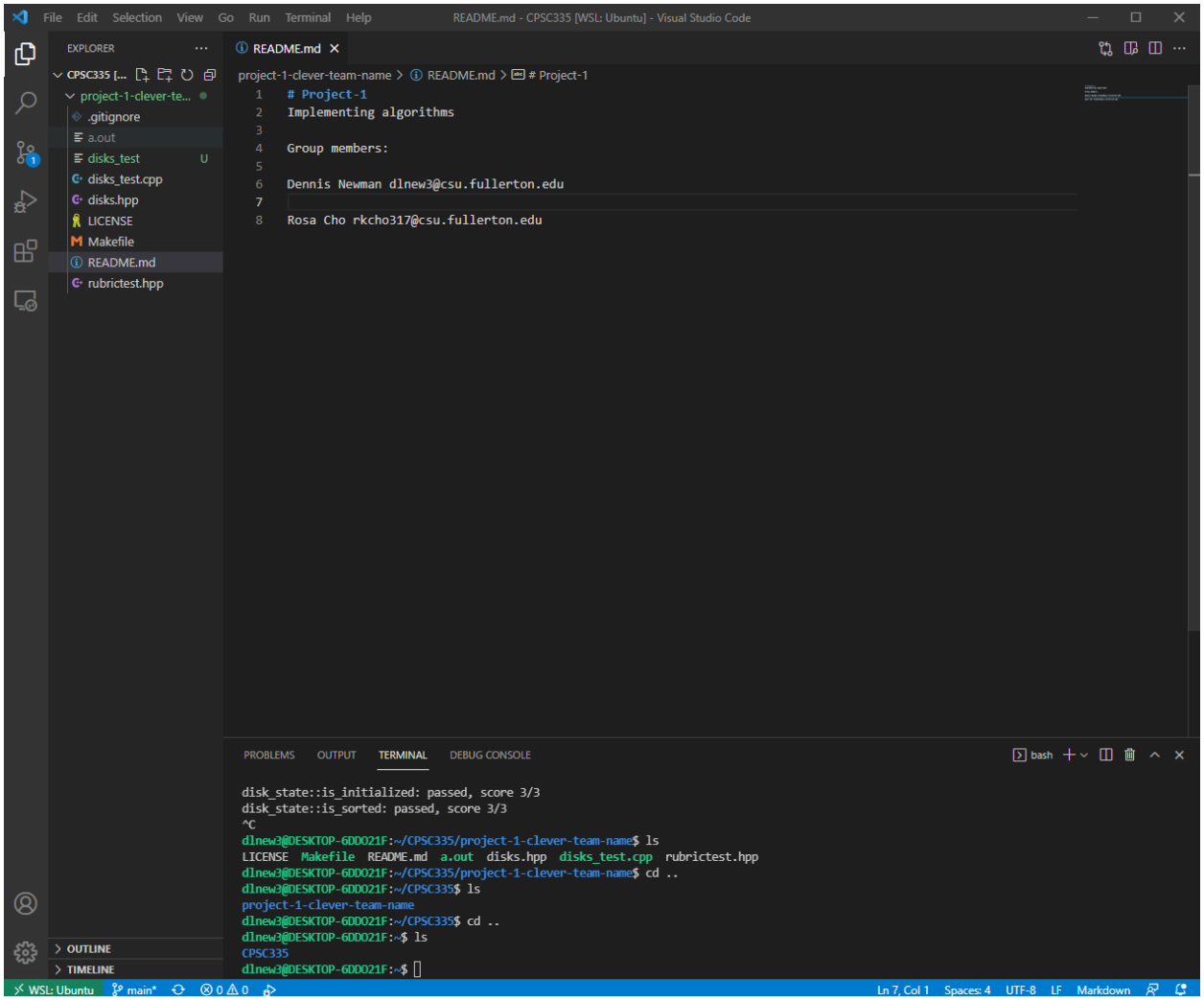
Dennis Newman - dlnew3@csu.fullerton.edu
Rosa Cho - rkcho317@csu.fullerton.edu

Project 1 Submission



Dennis Newman - dlnew3@csu.fullerton.edu
Rosa Cho - rkcho317@csu.fullerton.edu

```
dlnew3@DESKTOP-6DDO21F:~/CPSC335/project-1-clever-team-name$ make
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
In file included from disks_test.cpp:10:
disks.hpp: In member function ‘bool disk_state::is_initialized() const’:
disks.hpp:98:25: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
   98 |       for (int i = 0; i < total_count(); i++) {
      |                       ~~^~~~~~~~~~~~~~~
disks.hpp: In member function ‘bool disk_state::is_sorted() const’:
disks.hpp:116:25: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  116 |       for (int i = 0; i < total_count(); i++) {
      |                       ~~^~~~~~~~~~~~~~~
disks.hpp:118:17: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  118 |           if (i < light_count() && get(i) == DISK_LIGHT) {
      |               ~~^~~~~~~~~~~~~~
disks.hpp:123:22: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  123 |           else if (i >= dark_count() && get(i) == DISK_DARK) {
      |                    ~~^~~~~~~~~~~~~~
disks.hpp: In function ‘sorted_disks sort_alternate(const disk_state&)’:
disks.hpp:161:21: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  161 |   for (int i = 0; i < after.light_count(); i++){
      |                   ~~^~~~~~~~~~~~~~~~~~~~~
disks.hpp:163:25: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  163 |     for (int k = i; k+1 < after.total_count(); k+=2){
      |                     ~~~~^~~~~~~~~~~~~~~~~~~~~~
disks.hpp: In function ‘sorted_disks sort_lawnmower(const disk_state&)’:
disks.hpp:179:21: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  179 |   for (int i = 0; i < (after.light_count() / 2); i++){
      |                   ~~^~~~~~~~~~~~~~~~~~~~~~~~~~~
disks.hpp:181:23: warning: comparison of integer expressions of different signedness: ‘int’ and ‘size_t’ {aka ‘long unsigned int’} [-Wsign-compare]
  181 |     for (int j = 0; j < (after.total_count() - 2); j++){
      |                     ~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~
./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

dlnew3@DESKTOP-6DDO21F:~/CPSC335/project-1-clever-team-name$
```

Dennis Newman - dlnew3@csu.fullerton.edu
Rosa Cho - rkcho317@csu.fullerton.edu

Pseudocode

Lawnmower sort:

```
int swaps = 0;                                                  //1 step
for (int i = 0; i < disks.size() / 4; i++){                     //n/2 times
        for (int j = 0; j < disks.size() - 2; j++){             //n-1 times
                if (disks[j] == LIGHT && disks[j+1] == DARK){   //4 steps
                        temp = disks[j];                        //1 step
                        disks[j] = disks[j+1];                  //2 steps
                        disks[j+1] = temp;                      //2 steps
                        swaps++;                                //1 step
                }
        }
        for (int k = disks.size() - 1; k > 0; k--){             //n-1 times
                if (disks[k] == DARK && disks[k-1] == LIGHT){   //4 steps
                        temp = disks[k];                        //1 step
                        disks[k] = disks[k-1];                  //2 steps
                        disks[k-1] = temp;                      //2 steps
                        swaps++;                                //1 step
                }
        }
}
```

S.C. = $1 + (n/2 * ((n-1) * (4 + 1 + 2 + 2 + 1)) * ((n-1) * (4 + 1 + 2 + 2 + 1)))$
$= 1 + (10n^3 - 200n^2 + 100n)/2 = f(n)$

$f(n)$ is $O(n^3)$ if $f(n) <= cn^3$ for some $n >= n_0$:
Let $c = 156$ $(1 + 10/2 + 200/2 + 100/2)$. If $1 + (10n^3 - 200n^2 + 100n)/2 <= cn^3$ for some $n >= n_0$, then:

$1/n^3 + (10n^3 - 200n^2 + 100n)/2n^3 <= 156$

Or:

$1/n^3 + 5 - 100/n + 50/n^2 <= 156$, which is true for $n >= 1$.

Therefore, the Big-O condition holds for $n >= n_0 = 1$ and $c >= 156$.

Dennis Newman - dlnew3@csu.fullerton.edu
Rosa Cho - rkcho317@csu.fullerton.edu

Alternate Sort:

```
int swaps = 0;                                                  //1 step
for (int i = 0; i < disks.count() / 2; i++){                    //n times
        for (int k = i; k < disks.count()-1; k+=2){             //(n - 1 - i)/2 times
                if (disks[k] == LIGHT && disks[k+1] == DARK){ //4 steps
                        temp = disks[k];                        //1 step
                        disks[k] = disks[k+1];                  //2 step
                        disks[k+1] = temp;                      //2 step
                        swaps++;                                //1 step
                }
        }
}
```

S.C. = $1 + \Sigma_{(i=0 \rightarrow n)} * \Sigma_{(j=i \rightarrow (n-1)/2)} * 10$

$= 1 + (\Sigma_{(i=0 \rightarrow n)}(10n-10) - \Sigma_{(i=0 \rightarrow n)}(10i) - \Sigma_{(i=0 \rightarrow n)}(10))$
$= 1 + (\Sigma_{i=0 \rightarrow n)}((10n-10)(n+1) - \Sigma_{i=0 \rightarrow n)}(10n(n+1))/2 + \Sigma_{i=0 \rightarrow n)}10(n+1)))$
$= 1 + (10n^2 + 10n - 10n - 10) - (10n^2 + 10n)/2 + 10n+10)$
$= 1 + ((10n^2 - 10) - (5n^2 + 5n) + 10n + 10)$
$= 1 + (10n^2 - 5n^2 -5n + 10n +10 - 10)$
$= 1 + (5n^2 + 5n)$
$= 5n^2 + 5n + 1 = g(n)$

g(n) is $O(n^2)$ if $g(n) <= cn^2$ for some $n >= n_0$:
Let c = 11 (5 + 5 +1). If $5n^2 + 5n + 1 <= 11n^2$ for some $n >= n_0$, then:

$5n^2/n^2 + 5n/n^2 + 1/n^2 <= 11$
or
$5 + 5/n + 1/n^2 <= 11$, which is true for all $n >= 1$.

Therefore, the Big-O condition holds for $n >= n_0 = 1$ and $c >= 11$.