

Project 1 -Implementing Algorithms

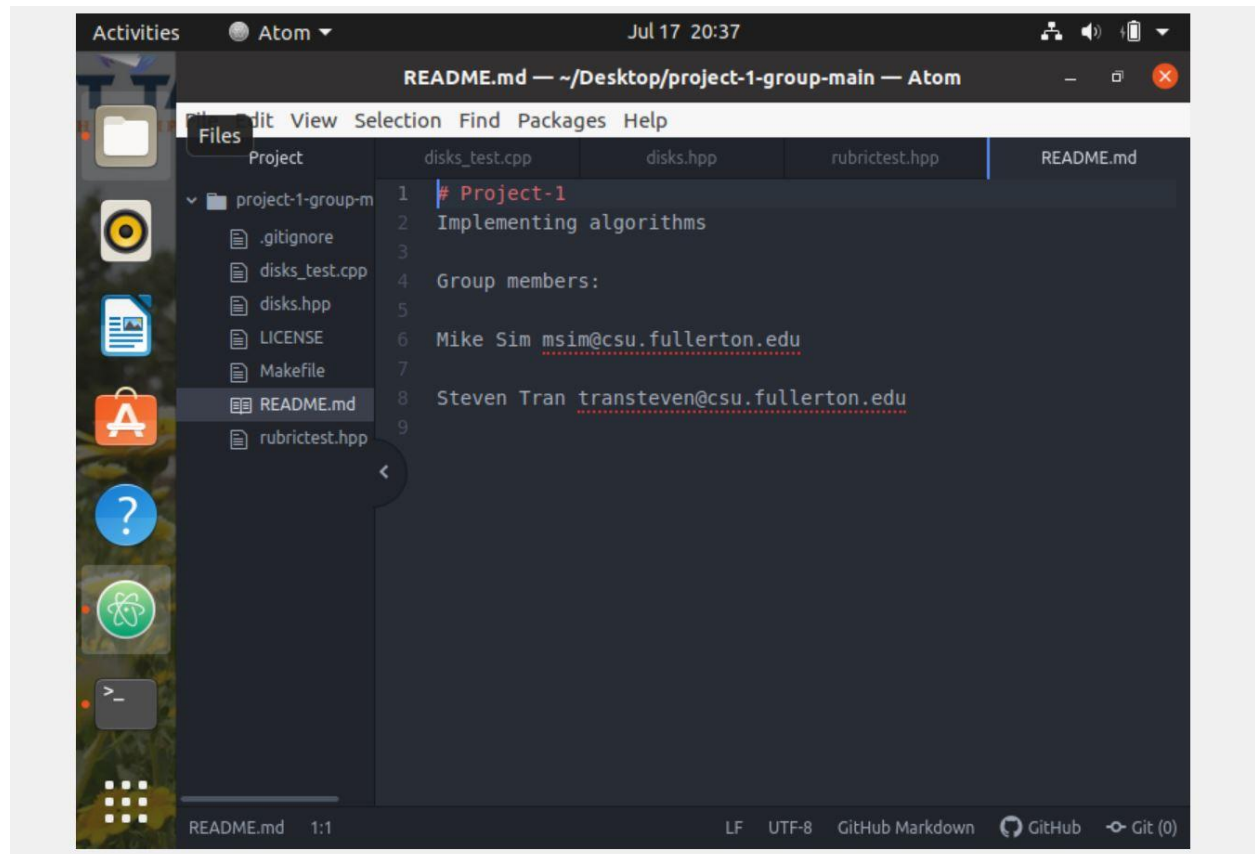
Mike Sim msim@csu.fullerton.edu

Steven Tran transteven@csu.fullerton.edu

CPSC 335-02 Algorithm Engineering

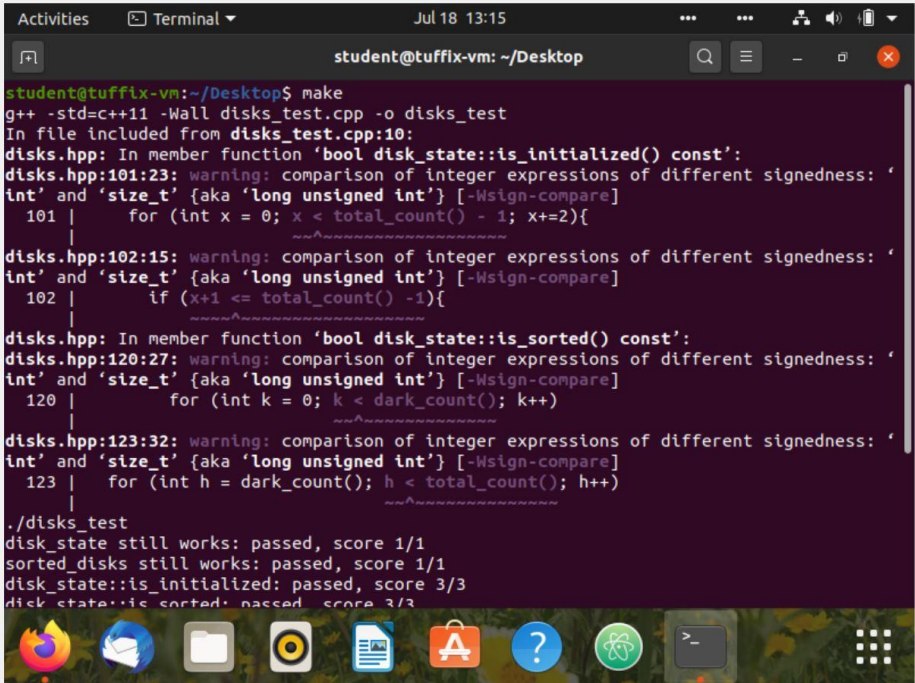
Date 18 July, 2021

Screenshot of group members within the Tuffix and Atom text editor.



Screenshots of running the makefile are below

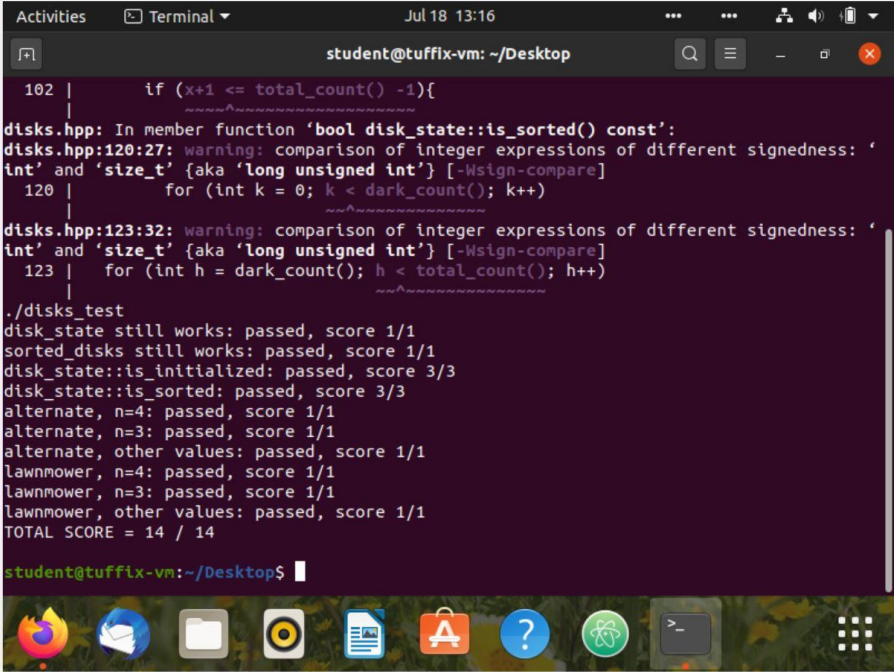
Tuffix 2020 Edition [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help



```
Activities Terminal Jul 18 13:15
student@tuffix-vm: ~/Desktop

student@tuffix-vm:~/Desktop$ 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:101:23: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
101 |     for (int x = 0; x < total_count() - 1; x+=2){
    |                      ^
disks.hpp:102:15: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
102 |         if (x+1 <= total_count() - 1){
    |             ^
disks.hpp: In member function 'bool disk_state::is_sorted() const':
disks.hpp:120:27: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
120 |         for (int k = 0; k < dark_count(); k++)
    |                      ^
disks.hpp:123:32: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
123 |     for (int h = dark_count(); h < total_count(); h++)
    |                                ^
./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
```

Tuffix 2020 Edition [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help



```
102 |         if (x+1 <= total_count() - 1){
    |                      ^
disks.hpp: In member function 'bool disk_state::is_sorted() const':
disks.hpp:120:27: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
120 |         for (int k = 0; k < dark_count(); k++)
    |                      ^
disks.hpp:123:32: warning: comparison of integer expressions of different signedness: '
int' and 'size_t' {aka 'long unsigned int'} [-Wsign-compare]
123 |     for (int h = dark_count(); h < total_count(); h++)
    |                                ^
./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$
```

Mathematical Analysis on Pseudocode is done below

Pseudocode Alternate Algorithm and Mathematical Analysis

for (size i = 0 to i < total_count() / 2;)	i=0 to n/2
{	
for (size j = i to j < total_count() - 1;)	j=i to n-1
{	
if ((j)>(j+1))	---2
{	
swap;	---0
count++;	---1
}	
}	
}	
return sorted_disks(before, count);	--returned

Step Count (if)=2+max(1,0)=3

$$\begin{aligned}
 \sum_{i=0}^{n/2} \sum_{j=i}^{n-1} 3 &= \sum_{i=0}^{n/2} 3((n-1)-i+1) = \sum_{i=0}^{n/2} 3n-3i \\
 &\quad \underbrace{\hspace{10em}}_{\text{dependent}} \\
 &= 3 \sum_{i=0}^{n/2} 1 - 3 \sum_{i=0}^{n/2} i = \frac{3n}{2} - 3 \left(\frac{\frac{n}{2}(\frac{n}{2}+1)}{2} \right) = \frac{3n}{2} - \frac{3n^2}{4} - \frac{6n}{4} \\
 &= \frac{3n^2}{2} - \frac{3n}{2} \in O\left(\frac{3n^2}{2} - \frac{3n}{2}\right) \\
 &\Rightarrow O\left(\frac{3}{2}n^2 - 3n\right) \\
 &= O(n^2) \text{ as constraining factor} \\
 \therefore \text{The alternative algorithm take } \underline{O(n^2)} \text{ times}
 \end{aligned}$$

Pseudocode Lawnmower Algorithm and Mathematical Analysis

```
for(size i=0 to i<total_count()/2;)          ---i=0 to n/2
{
    for (size j=i to total_count()-1;)        ---j=i to n-1
    {
        if( i>i+1)                            ---2
        {
            swap(i);                          ---0
            Count++;                          ---1
        }
    }
}
for(size j= total_count()-2, j>i)            ---j=n-2, j is larger than i
{
    if(j<j-1)                                  ---2
    {
        swap(j-1)                            ---1
        Count++;                            ---1
    }
}
}
return sorted_disks(before, count);          ---return
```

First if step $\text{count}=2+\max(0,1)=3$

Second if step $\text{count}=2+\max(1,1)=3$

$$\begin{aligned}
& \underbrace{\sum_{i=0}^{n/2} \sum_{j=i}^{n-1} 3 + \sum_j^{n-2} 3}_{\text{dependent}} \\
&= \sum_{i=0}^{n/2} 3((n-i)-i+1) + 3((n-i)-i+1) \\
&= \sum_{i=0}^{n/2} 3n - \sum_{i=0}^{n/2} 3i + (3n - 3i - 3) \\
&\hookrightarrow \frac{3n}{2} - 3\left(\frac{n/2(n/2+1)}{2}\right) + (3n - 3i - 3) \\
&3\left(\frac{\frac{n^2}{4} + \frac{n}{2}}{2}\right) = \left(\frac{3n^2}{2}\right) + 3n + \frac{9n}{2} \\
&\hookrightarrow \frac{3}{2}n^2 \in \underline{O(n^2)}
\end{aligned}$$

\therefore the lawnmower algorithm takes $O(n^2)$