

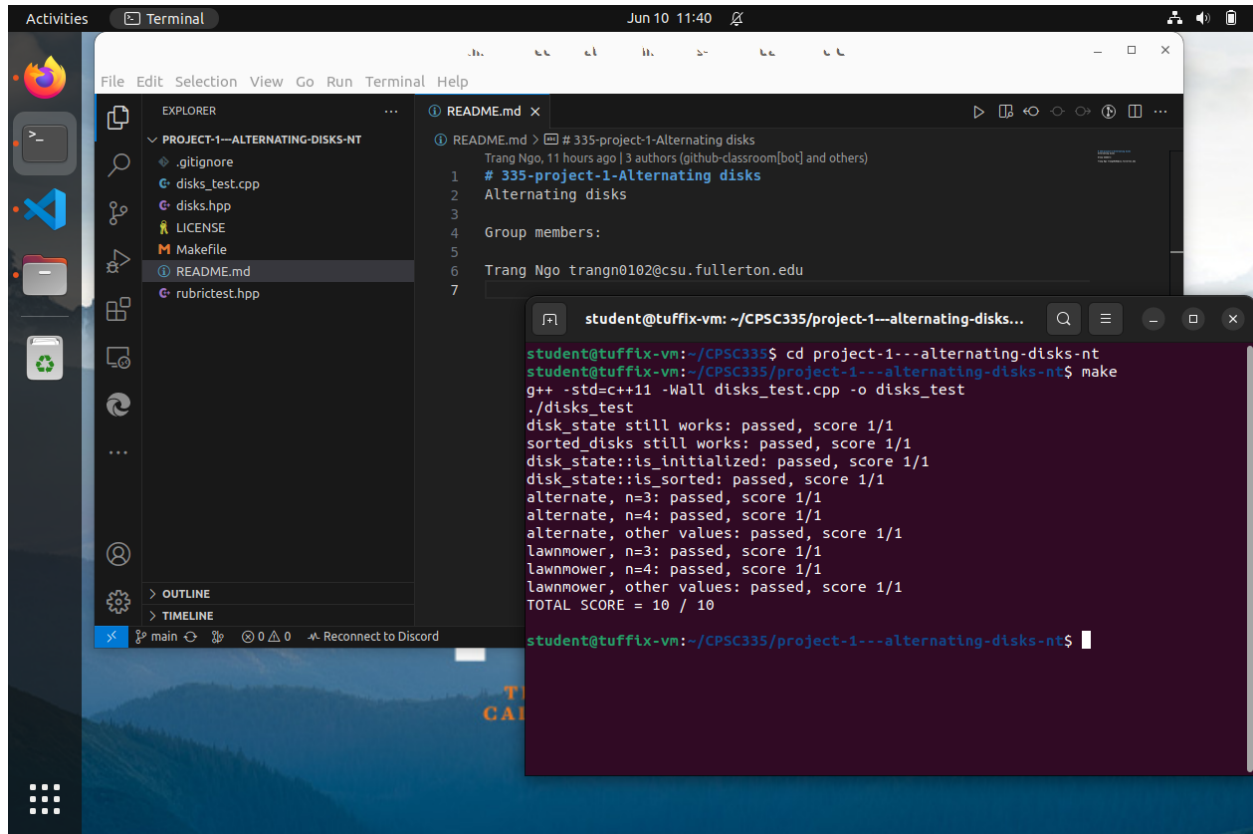
Project 1 Report

Group member

Trang Ngo trangn0102@csu.fullerton.edu

This is my submission for Project 1: Alternating Disks

Tuffix Screenshot:



The screenshot displays a Tuffix VM environment. On the left, a file explorer shows the project structure for 'PROJECT-1--ALTERNATING-DISKS-NT', including files like .gitignore, disks_test.cpp, disks.hpp, LICENSE, Makefile, README.md, and rubric_test.hpp. The main editor window shows the README.md file, which contains the following text:

```
1 # 335-project-1-Alternating disks
2 Alternating disks
3
4 Group members:
5
6 Trang Ngo trangn0102@csu.fullerton.edu
7
```

In the foreground, a terminal window shows the execution of the program. The user navigates to the project directory and runs the following commands:

```
student@tuffix-vm: ~/CPSC335/project-1--alternating-disks-nt
student@tuffix-vm: ~/CPSC335/project-1--alternating-disks-nt$ make
g++ -std=c++11 -Wall disks_test.cpp -o disks_test
./disks_test
disk_state still works: passed, score 1/1
sorted_disks still works: passed, score 1/1
disk_state::is_initialized: passed, score 1/1
disk_state::is_sorted: passed, score 1/1
alternate, n=3: passed, score 1/1
alternate, n=4: passed, score 1/1
alternate, other values: passed, score 1/1
lawnmower, n=3: passed, score 1/1
lawnmower, n=4: passed, score 1/1
lawnmower, other values: passed, score 1/1
TOTAL SCORE = 10 / 10
student@tuffix-vm: ~/CPSC335/project-1--alternating-disks-nt$
```

Pseudocode

Pseudocode

Lawnmower algorithms

```
def lawnmower (disk)
  disk = total length of disk // 1
  swap-count = 0 // 1
  for i from 0 to  $\frac{n+1}{2}$  do //  $(\frac{n+1}{2} - 0) + 1 =$ 
    for a from 0 to  $2n-2$  do //  $2n-2-0+1 = 2n-1$ 
      if (disk at index a == light and disk at index a+1 == dark) // 3
        swap (disk at index a, disk at index a+1) // 1
        increment of swap-count by 1 // 1
      endif // 0
    endfor // 0
  for b from  $2n-1$  down to 1 do //  $((\frac{1-(2n-1)}{2}) + 1) = (-1(1-(2n-1)) + 1) = 2n-2+1 = 2n-1$ 
    if (disk at index b == dark and disk at index b-1 == light) // 3
      swap (disk at index b, disk at index b-1) // 1
      increment of swap-count by 1 // 1
    endif // 0
  endfor // 0
endfor // 0
return lawnmower of the disk and swap-count // 0
```

Step count

$$SC = 1 + 1 + \left(\frac{n+1}{2} + 1 \right) * [(2n-1) * (3 + \max(1,1))] + [(2n-1) * 3 + \max(1,1)]$$

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

$$= 2 + \frac{n+3}{2} * [8n-4 + 8n-4]$$

$$= 2 + \frac{n+3}{2} * [16n - 8]$$

$$= 2 + \frac{(16n^2 - 8n + 48n - 24)}{2}$$

$$= 2 + \frac{(16n^2 + 40n - 24)}{2}$$

$$= 2 + 8n^2 + 20n - 12$$

$$= 8n^2 + 20n - 10$$

$$\text{Time complexity} = O(n^2)$$

Proof 1 : $8n^2 + 20n - 10 \in O(n^2)$

By definition, you need to find $c > 0$ and $n_0 \geq 0$ such that

$$8n^2 + 20n - 10 \leq c * n^2$$

Choose $c = |8| + |20| + |-10| = 38$

So $8n^2 + 20n - 10 \leq 38n^2 \quad \forall n \geq n_0$
and $n_0 = 0$

Proof 2 : $8n^2 + 20n - 10 \in O(n^2)$ by limit theorem

$$\lim_{n \rightarrow \infty} \frac{8n^2 + 20n - 10}{n^2}$$

$$= \lim_{n \rightarrow \infty} \frac{(8n^2 + 20n - 10)'}{(n^2)'}$$

$$= \lim_{n \rightarrow \infty} \frac{16n + 20}{2n}$$

$$= \lim_{n \rightarrow \infty} \frac{(16n + 20)'}{(2n)'}$$

$$= \lim_{n \rightarrow \infty} \frac{16}{2}$$

$= 8 \geq 0$ and a constant

conclude $8n^2 + 20n - 10 \in O(n^2)$

Alternate algorithm

```
def alternate ( disk )
  disk = total length of disk // 1
  swap_count = 0 // 1
  for i = 0 to n // (n-0)+1 = n+1
    if (i % 2 == 0) do // 2
      // Start with leftmost disk
      for a from 0 to 2n-1 skip 2 do // ((2n-1-0)/2)+1 = (2n-1)/2 + 1
        if (disk at index a == light and disk at index a+1 == dark) // 3
          swap (disk at index a, disk at index a+1) // 1
          increment of swap_count by 1 // 1
        endif // 0
      endfor // 0
    else
      // Start with second leftmost disk
      for b from 1 to 2n-2 skip 2 do // ((2n-2-1)/2)+1 = (2n-3)/2 + 1
        if (disk at index b == light and disk at index b+1 == dark) // 3
          swap (disk at index b, disk at index b+1) // 1
          increment of swap_count by 1 // 1
        endif // 0
      endfor // 0
    endfor // 0
  endfor // 0
  return alternate of the disk and swap-count // 0
```

Step Count

$$S.C = 1 + 1 + (n+1) * \left[2 + \max \left(\left(\frac{2n-1}{2} + 1 \right) * (3 + \max(1,1)), \right. \right. \\ \left. \left. \left(\frac{2n-3}{2} + 1 \right) * (3 + \max(1,1)) \right) \right]$$

$$= 2 + (n+1) * \left[2 + \max \left(\left(\frac{2n-1}{2} + \frac{2}{2} \right) * 4, \left(\frac{2n-3}{2} + \frac{2}{2} \right) * 4 \right) \right]$$

$$= 2 + (n+1) * \left[2 + \max \left(\left(\frac{2n+1}{2} \right) * 4, \left(\frac{2n-1}{2} \right) * 4 \right) \right]$$

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

$$= 2 + (n+1) * \left[2 + \max(4n+2, 4n-2) \right]$$

$$= 2 + (n+1) * [2 + 4n+2]$$

$$= 2 + (n+1) * [4n+4]$$

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

$$= 2 + 4n^2 + 8n + 4$$

$$= 4n^2 + 8n + 6$$

$$\text{Time Complexity} = O(n^2)$$

Proof 1 : $4n^2 + 8n + 6 \in O(n^2)$

By definition, you need to find $c > 0$ and $n_0 \geq 0$ such that

$$4n^2 + 8n + 6 \leq c * n^2$$

Choose $c = |4| + |8| + |6| = 18$

$$\text{So } 4n^2 + 8n + 6 \leq 18n^2 \quad \forall n \geq n_0$$

and $n_0 = 0$

Proof 2 : $4n^2 + 8n + 6 \in O(n^2)$ by limit theorem

$$\lim_{n \rightarrow \infty} \frac{4n^2 + 8n + 6}{n^2}$$

$$= \lim_{n \rightarrow \infty} \frac{(4n^2 + 8n + 6)'}{(n^2)'}$$

$$= \lim_{n \rightarrow \infty} \frac{8n + 8}{2n}$$

$$= \lim_{n \rightarrow \infty} \frac{(8n + 8)'}{(2n)'}$$

$$= \lim_{n \rightarrow \infty} \frac{8}{2}$$

$$= 4 \geq 0 \text{ and a constant}$$

Conclude $4n^2 + 8n + 6 \in O(n^2)$