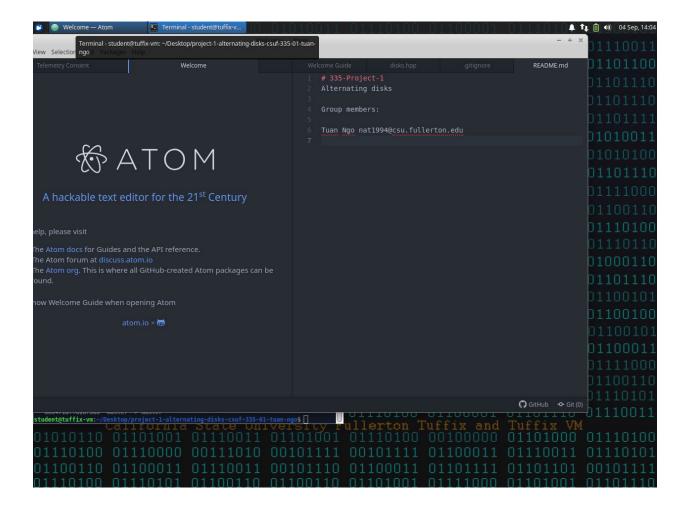
Name: Tuan Ngo

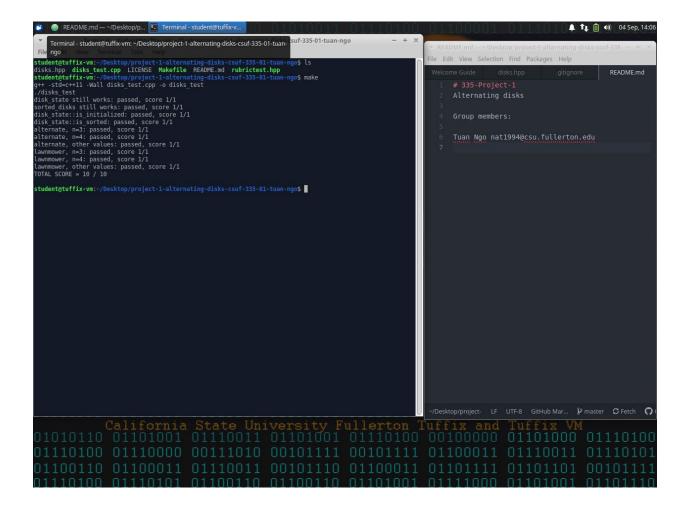
CSUF email: nat1994@csu.fullerton.edu

CWID: 887416766

CPSC 335 Fall 2020

Project 1: Alternating Disks





Pseudocode:

1. sort_alternate

```
sorted_disks sorted_disks sort_alternate(const disk_state& before):
        count = 0
                                                                                          // 1 t.u
        tmp disks = before
                                                                                          // 1 t.u
        for i = 0 to size(tmp_disks) - 1 do:
                                                                                 //analysis later
                for j = i + 1 to size(tmp_disks) - 1 do:
                                                                         //analysis later
                        if tmp_disks[j - 1]==DISK_LIGHT && tmp_disks[j] ==DISK_DARK: // 3 t.u
                                                                         // Professor told 3 t.u.
                                        tmp_disks.swap(j - 1)
                                        count++
                                                                                          // 1 t.u
                        endif
                endfor
        endfor
        return sorted_disks(tmp_disks, count)
```

analysis:

assume:

size(Disks) = n
return does not count time unit
lookup array/vector/list does not count time units

s.c =
$$2 + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} (3 + \max(0,4))$$

s.c = $2 + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} 7$
s.c = $2 + \sum_{i=0}^{n-1} \sum_{j=i}^{n-2} 7$

Calculate:
$$\sum_{j=i}^{n-2} 7 = \sum_{j=1}^{n-2} 7 - \sum_{j=1}^{i-1} 7$$

= $7(n-2) - 7(i-1)$
= $7(n-2-i+1)$
= $7(n-i-1)$

s.c =
$$2 + \sum_{i=0}^{n-1} 7(n-i-1)$$

= $2 + \sum_{i=0}^{n-1} 7(n-1) - \sum_{i=0}^{n-1} 7i$
= $2 + 7(n-1)(n-1) - 7\frac{n(n-1)}{2}$
= $2 + 7(n^2 - 2n + 1) - \frac{7n^2 - 7}{2}$
= $(4 + 14n^2 - 28n + 14 - 7n^2 + 7)/2$
s.c = $(7n^2 - 28n + 25)/2$

Time complexity: O(n²)

Prove:

Show ($(7n^2 - 28n + 25)/2$) belongs to $O(n^2)$

Find c >= 0 and n_0 >= 0 such that $4n^2$ - 2 <= c* n^2 withall n>= n_0 Choose c = 60/2 = 30 ($7n^2$ - 28n + 25) /2 <= $60n^2/2$ $7n^2$ - 28n + 25 <= $60n^2$ $7n^2$ - 28n + 25 <= $7n^2$ + $28n^2$ +

```
sorted_disks sort_lawnmower(const disk_state& before):
          tmp disk = before;
                                                                                                            // 1 t.u
                                                                                                            // 1 t.u
          count = 0
          for i = 0 to (size(Disks)/2) - 1 do:
                                                                              //Analysis later
                    for j = 1 + i to size(Disks) - 1 do:
                                                                    //Analysis later
                              if tmp_disks[j-1] == DISK_LIGHT && tmp_disks[j] == DISK_DARK: // 3 t.u
                                       then:
                                                 tmp_disks.swap(j - 1) // Professor told 3 t.u.
                                                  count++;
                                                                                                            // 1 t.u
                                       endthen
                              endif
                    endfor
                    for k = size(Disks) - 1 - i down to 1 do:
                              if tmp_disks[k] == DISK_DARK && tmp_disks[k - 1]==DISK_LIGHT: //3 t.u
                                       then:
                                                                                       // Professor told 3 t.u.
                                                 tmp_disks.swap(k - 1)
                                                 count++;
                                                                                                            // 1 t.u
                                       endthen
                              endif
                    endfor
          endfor
          return sorted_disks(tmp_disks,count)
 analysis:
          assume:
                    size(Disks) = n
                    return does not count time unit
                    lookup array/vector/list does not count time units
s.c = 2 + \sum_{i=0}^{\frac{n}{2}-1} \left( \sum_{j=i+1}^{n-1} (3 + \max(0,4)) + \sum_{k=1}^{n-1} (3 + \max(0,4)) \right)
s.c = 2 + \sum_{i=0}^{\frac{n}{2}-1} \sum_{j=i+1}^{n-1} (3 + \max(0,4)) + \sum_{i=0}^{\frac{n}{2}-1} \sum_{k=1}^{n-1-i} (3 + \max(0,4))
```

Time complexity: O(n²)

Prove:

Show ($(21n^2 + 42n + 8)/4$) belongs to O(n^2)

Find c>=0 and n_0 >=0 such that $6n^2 + 12n + 2 <= c* n^2$ withall n>= n_0 Choose c= 71/4 ($21n^2 + 42n + 8$) /4 <= $71n^2/4$ 2 $1n^2 + 42n + 8$ <= $21n^2 + 42n^2 + 8n^2$ 2 $1n^2 - 6n^2 + 42n^2 - 42n + 8n^2 - 8$ >= 0 n>= 1 and n_0 = 1