> forkbomb

Patrick Roteman

Joshua Filstrup

Tim Stullich

Assignment #5

--

###Solutions:

\*\*FCFS\*\*:

First Come First Serve simply iterates through the list of positions taking the distance between the previous and next

and adding it to the total distance traveled

Total distance traveled:

|(2255-2055)| + | (2055-1175) | + | (1175-2034) | + | (2034-2700) | + | (2700-513) | + | (513-1680) | +

| (1680-256) | + | (256-1401) | + | (1401-4922) | + | (4922-3692) |

= 13279

Order of cylinders visited:

2255,2055,1175,2304,2700,513,1680,256,1401,4922,3692

\*\*SSTF\*\*

Shortest Seek Time First takes the minimum element in the queue from the current head position and moves the head there.

While this greedy algorithm may produce good results, it is not guaranteed to produce a near optimal solution.

Total distance traveled:

| ( 2255 - 2304 )| + | ( 2304 -2055 )| + | ( 2055 - 1680 )| +| ( 1680 - 1401 )|

+ | ( 1401 - 1175 )| + | ( 1175 - 513 )| + | ( 513 - 256 )| + | ( 256 - 2700 )|

+ | ( 2700 - 3692 )| + | ( 3692 - 4922 )|

= 6763

Order of cylinders visited:

2255,2304,2055,1680,1401,1175,513,256,2700,3692,4922

\*\*Look\*\*:

Look will go up, reading everything until the max, then read everything else on the way back

Total distance traveled:

(4922 - 2255) + (4922 - 256)

= 7333

Order of cylinders visited:

2255,2304,2700,3692,4922,2055,1680,1401,1175,513,256

\*\*C-Look\*\*:

CLook will go up, reading everything until the max, then go all the way to minimum, then read until it gets the rest

Total distance traveled:

(4922 - 2255) + (4922 - 256) + (2055 - 256)

= 9132

Order of cylinders visited:

2255,2304,2700,3692,4922,256,513,1175,1401,1680,2055

\*\*Scan\*\*:

Scan will go up, reading everything until 4999, then read everything else on the way back

Total distance traveled:

(4999 - 2255) + (4999 - 256)

= 7487

Order of cylinders visited:

2255,2304,2700,3692,4922,2055,1680,1401,1175,513,256

\*\*C-Scan\*\*:

C-Scan will go up, reading everything until the max, then go all the way to 0, then read until it gets the rest

Total distance traveled:

(4999 - 2255) + (4999 - 0) + (2055 - 0)

= 9798

Order of cylinders visited:

2255,2304,2700,3692,4922,256,513,1175,1401,1680,2055

###Code:

We wrote the following python code to quickly verify our computations:

```python

## hdd.py

## Joshua Filstrup

## Written for CS 149, SJSU, Mak

import math

headstart = 2255

max\_read\_pos = 4999

def fcfs(l):

total\_dist = 0

for i in range(0,len(l)-1):

total\_dist += abs(l[i+1] - l[i])

return total\_dist

def sstf(l):

total\_dist = 0

curr = l[0]

l.remove(headstart)

while len(l) > 0:

min\_dist = abs(curr - l[0])

next\_elt = l[0]

for i in range(1, len(l)):

if abs(curr - l[i]) < min\_dist:

min\_dist = abs(curr - l[i])

next\_elt = l[i]

l.remove(next\_elt)

curr = next\_elt

total\_dist += min\_dist

return total\_dist

def look(l):

max\_elt = max(l)

total\_dist = 0

curr = l[0]

l.remove(headstart)

while len(l) > 0 and curr <= max\_elt:

curr += 1

total\_dist += 1

if curr in l:

l.remove(curr)

while len(l) > 0 and curr >= 0:

curr -= 1

total\_dist += 1

if curr in l:

l.remove(curr)

return total\_dist

def clook(l):

max\_elt = max(l)

total\_dist = 0

curr = l[0]

l.remove(headstart)

while len(l) > 0:

curr += 1

total\_dist += 1

if curr > max\_elt:

total\_dist += (curr-min(l))

curr = min(l)

if curr in l:

l.remove(curr)

return total\_dist

def scan(l):

total\_dist = 0

curr = l[0]

l.remove(headstart)

while len(l) > 0 and curr <= max\_read\_pos:

curr += 1

total\_dist += 1

if curr in l:

l.remove(curr)

while len(l) > 0 and curr >= 0:

curr -= 1

total\_dist += 1

if curr in l:

l.remove(curr)

return total\_dist

def cscan(l):

total\_dist = 0

curr = l[0]

l.remove(headstart)

while len(l) > 0:

curr += 1

total\_dist += 1

if curr >= max\_read\_pos:

curr = 0

total\_dist += max\_read\_pos

if curr in l:

l.remove(curr)

return total\_dist

```