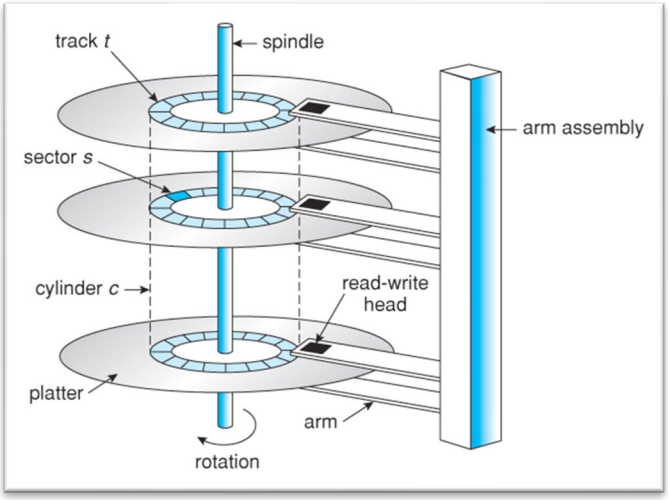
# How to Find the Total Seek Time of Disk Scheduling Algorithms

## Introduction

Disk scheduling is an operating system concept. For this tutorial I will be referencing a textbook named *Operating System Concepts.* Before we start the tutorial, I would like to review some terminology. I have also included a picture to showcase the internal parts of a hard disk drive. Feel free to skip the introduction if you are already familiar.

| Disk Scheduling Terminology | |
|---|---|
| **Bandwidth** | The total amount of data transferred divided by the total time between the first request for service and the completion of the last transfer. |
| **Cylinder** | On an HDD, the set of tracks under the read-write heads on all platters in the device. |
| **Disk Arm** | An HDD component that holds the read/write head and moves over cylinders of platters |
| **Disk Scheduling** | An operating system service that schedules data I/O requests of a hard drive. |
| **Disk Scheduling Algorithms** | A collection of scheduling algorithms with varying levels of fairness, timeliness, and optimization. |
| **Hard Disk Drive (HDD)** | A secondary storage device based on mechanical components, including spinning magnetic media platters, and moving read-write heads. |
| **I/O** | Input/Output |
| **Operating System** | A program that manages a computer's hardware, provides a basis for application programs, and acts as an intermediary between the computer user and the computer hardware. |
| **Platter** | An HDD component that has a magnetic media layer for holding charges. |
| **Seek Time** | The time it takes to locate the read/write data on the hard disk drive. |

*Source: Operating System Concepts Tenth Edition, Silberschatz et al. (2018)*




HDD moving-head disk mechanism

*Source: Operating System Concepts Tenth Edition, Silberschatz et al. (2018)*

# Purpose & Example Prompt

Disk scheduling algorithms are used to schedule input/output requests of a hard disk drive. Disk scheduling algorithms aim to decrease seek time and increase data transfer bandwidth. The differing algorithms represent different scheduling orders.

In this tutorial, we will use the same list of I/O requests to find the total seek time of 3 different disk scheduling algorithms.

```
Use Disk Scheduling to find the total seek time for the following I/O requests:
                                  25, 40, 17, 10, 58.
Start at 30.
The disk cylinder range is 200 [0 to 199].
```

Here are the three scheduling algorithms we will be covering:

- FCFS Scheduling
- SCAN Scheduling
- C-SCAN Scheduling (moving to the right)

# FCFS Scheduling

First come first serve scheduling is very easy to schedule. This algorithm is not very fast, but it is fair in the way it accesses data.

To find the total seek time, we will need to know the initial head position. Consider the I/O request list:

25, 40, 17, 10, 58

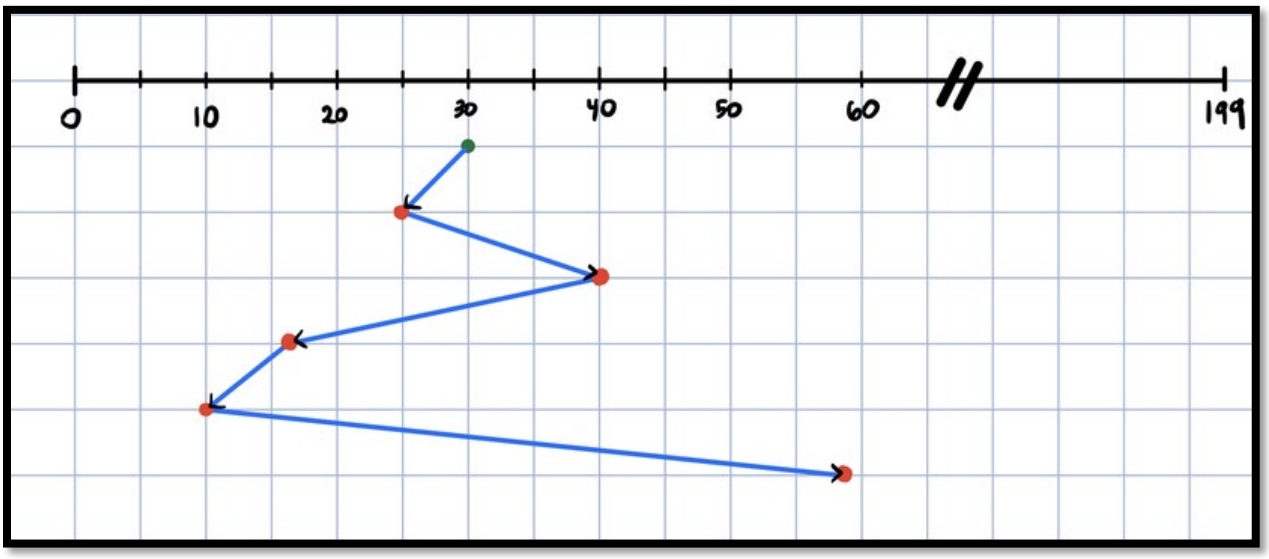
The initial head position is 30 and the range is 0 to 199.

## Steps to schedule the I/O requests using FCFS

**Step 1** Start at initial head position.

*The initial head position is 30.*

**Step 2** Move to the next I/O request in the list and repeat until the list is done.




*FCFS Scheduling Example*

**Step 3** Calculate total seek time by adding all the distances up from position to position (e.g., each line segment from the image above).

(30 - 25) + (40 - 25) + (40 - 17) + (17 - 10) + (58 - 10)

5 + 15 + 23 + 7 + 48

**98**

**Total Seek Time for FCFS Scheduling is 98**

# SCAN Scheduling

SCAN scheduling acts just like an elevator. The disk arm services I/O requests only in the direction that it is initially moving towards (left or right). When the disk arm reaches the extreme end of the disk, the head movement is reversed, and servicing is continued the other way. Basically, the head continuously scans back and forth across the disk.

To find the total seek time, we will need to know the initial head position and the direction of the head movement. Consider the I/O request list:

25, 40, 17, 10, 58

The initial head position is 30 and the range is 0 to 199.

## Steps to schedule the I/O requests using SCAN

**Step 1** Start at initial head position

*The initial head position is 30.*

**Step 2** Figure out the direction of the head movement.

*The direction is to the left (30 to 25; toward 0).*

This is easily discovered by looking at the next immediate position in the list.

- If the next I/O request is < the initial head position, then our direction will be to the left *(toward 0)*.
- If the next I/O request is > the initial head position, then the direction will be to the right *(toward 199)*.

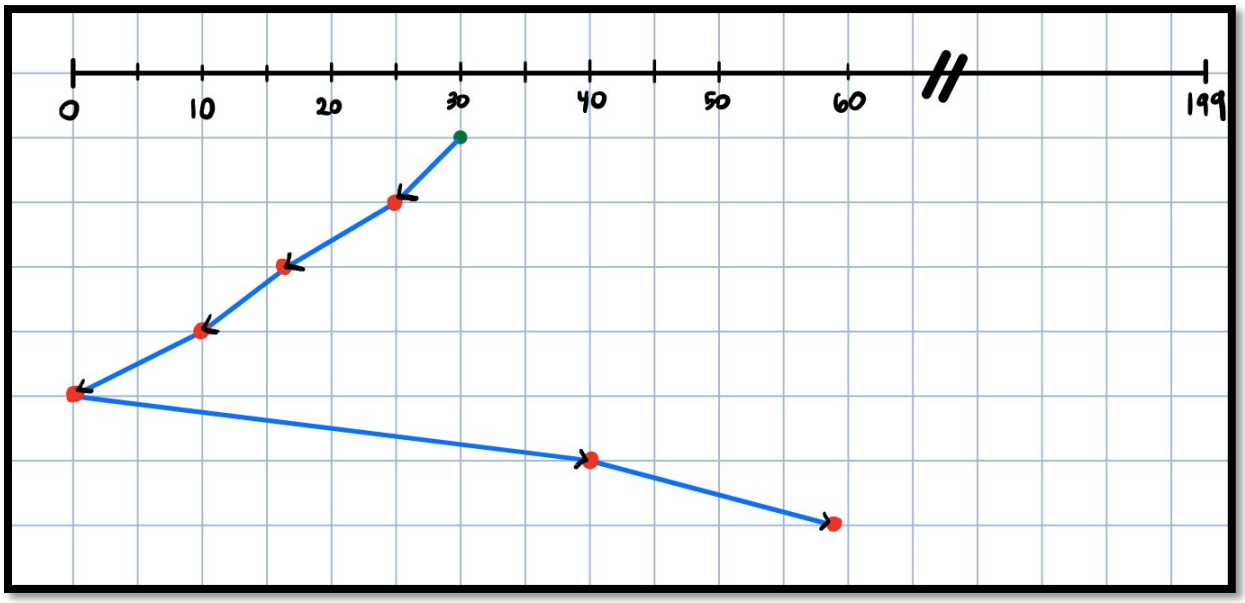
**Step 3** Service only the requests in the direction of the head movement (found in step 2).

**Step 4** Once all I/O requests have been completed in the direction traveled, go to the very end of the disk range.

*The extreme ends are 0 and 199.*

**Step 5** Reverse the head movement

**Step 6** Repeat Steps 3 to 5 until all I/O requests are serviced. Skip Steps 4 and 5 for the last I/O request.




*SCAN Scheduling Example*

**Step 7** Calculate total seek time by adding all the distances up from position to position (e.g., each line segment from the image above).

(30 - 25) + (25 - 17) + (17 - 10) + (10 - 0) + (40 - 0) + (58 - 40)

5 + 8 + 7 + 10 + 40 + 18

**88**

## OR

(30 - 0) + (58 - 0)

30 + 58

**88**

**Total Seek Time for SCAN Scheduling is 88**

# C-SCAN Scheduling

Circular SCAN (C-SCAN) scheduling is designed to provide a more uniform wait time than SCAN. Scheduling is exactly like SCAN, except that (I/O request) servicing is limited to only one direction. The disk arm services I/O requests only in the direction that it is initially moving towards (left or right). When the disk arm reaches the extreme end of the disk, it immediately returns to the other extreme end; without servicing any requests on the return trip. Basically, the head continuously scans back and forth across the disk, but only services requests in one direction.

To find the total seek time, we will need to know the initial head position. For C-SCAN, the head movement in which requests are scheduled is given. Consider the I/O request list:

25, 40, 17, 10, 58

The initial head position is 30 and the range is 0 to 199.

## Steps to schedule the I/O requests using C-SCAN (moving to the right)

**Step 1** Start at initial head position

*The initial head position is 30.*

**Step 2** Service only the requests in the direction of the given head movement.

*The servicing direction is to the right.*

Note: the servicing direction is given by the question prompt.

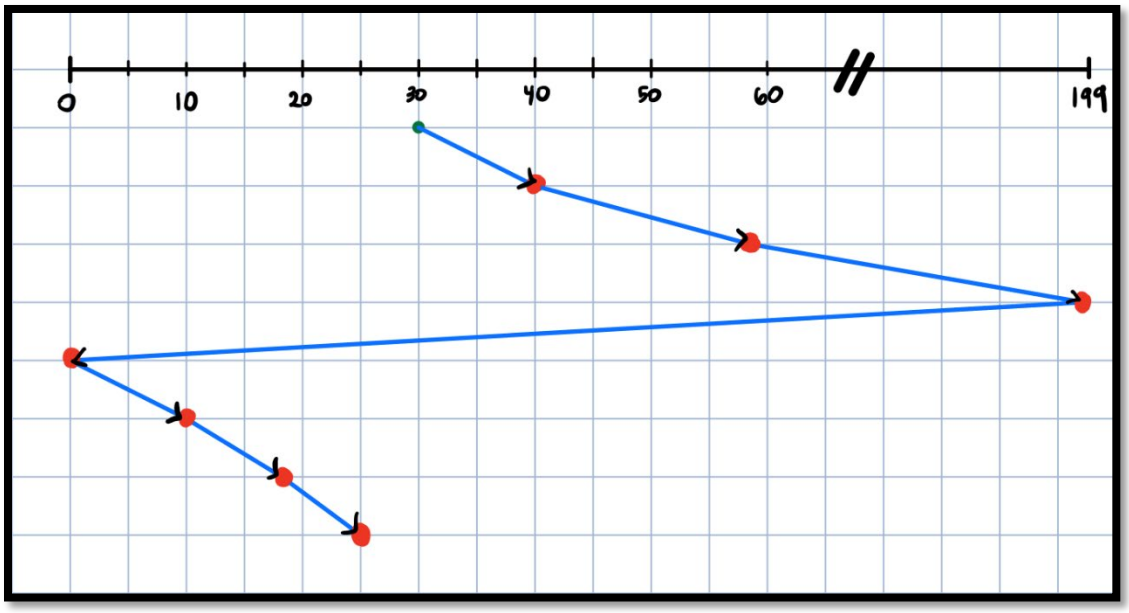
**Step 3** Once all I/O requests have been completed in the direction traveled, go to the very end of the disk range.

*The extreme ends are 0 and 199.*

**Step 5** Reverse the head movement and travel back to the other extreme end of the disk.

*The extreme ends are 0 and 199.*

**Step 6** Repeat Steps 2 to 5 until all I/O requests are serviced. Skip Steps 3 to 5 for the last I/O request.




*C-SCAN Scheduling Example*

**Step 7** Calculate total seek time by adding all the distances up from position to position (e.g., each line segment from the image above).

(40 - 30) + (58 - 40) + (199 – 58) + (199 - 0) + (10 - 0) + (17 - 10) + (25 - 17)

10 + 18 + 141 + 199 + 10 + 7 + 8

**393**

## OR

(199 - 30) + (199 - 0) + (25 - 0)

169 + 199 + 25

**393**

**Total Seek Time for C-SCAN Scheduling is 393**

## Results & Reference

Here are the seek times of each disk scheduling algorithm.

| | |
|---|---|
| FCFS | 98 |
| SCAN | 88 |
| C-SCAN | 393 |

If you would like to learn more on mass storage in general, here is the textbook resource used for this tutorial.

Silberschatz, Abraham, et al. "Operating System Concepts." *Operating System Concepts*, 10th Edition, Hoboken, NJ-United States, United States, Wiley, 2018, pp. 449–85.