Project Report



|  |  |
| --- | --- |
| Course Instructor: | **Sir Wakeel** |
| Total Marks: |  |
| Issue Date: | 20-11-2023 |
| Submission Date: | 08-01-2024 |

|  |  |  |
| --- | --- | --- |
| Submitted by: | Registration No. | Name |
| 21-CS-49 | Ehsan Ellahi |
|  | 21-CS-50 | Muhammad Burhan |
|  | 21-CS-64 | Muhammad Abdur Rehman |
| Obtained Marks: |  | |

**Project Report:** Comparative Analysis of N-Step Scan, F-Scan, and RSS Scan in Disk Scheduling

**1. Introduction:**

In the domain of disk scheduling techniques, N-Step Scan, F-Scan, and RSS Scan represent three methodologies crucial for optimizing data access and retrieval. This project report aims to provide an in-depth understanding of these scanning techniques, explore their applications in the context of disk scheduling, and highlight the differences between them.

**2. N-Step Scan and Disk Scheduling:**

N-Step Scan in the context of disk scheduling involves dividing the disk into N sectors and sequentially scanning each sector. The granularity of the scan is determined by the number of steps (N). Applications of N-Step Scan in disk scheduling include:

Sequential Access: N-Step Scan ensures systematic access to data on the disk, minimizing seek time and improving data retrieval efficiency.

Optimization for Image Storage: In scenarios where images or files are stored sequentially, N-Step Scan aligns with the access pattern, reducing access times.

**3. F-Scan and Disk Scheduling:**

F-Scan, emphasizing frequency analysis, correlates with disk scheduling algorithms that prioritize specific data frequencies or access patterns. Applications of F-Scan in disk scheduling include:

Frequent Data Access: F-Scan algorithms optimize for frequently accessed data, reducing seek time and enhancing overall disk performance.

Audio or Video File Access: In scenarios where certain frequencies represent critical audio or video data, F-Scan can efficiently organize data access.

**4. RSS Scan and Disk Scheduling:**

RSS Scan, focused on signal strength, finds relevance in disk scheduling by considering the physical location of data on the disk. Applications of RSS Scan in disk scheduling include:

Optimizing Wireless Disk Access: In scenarios where disks are part of wireless storage systems, RSS Scan helps optimize access based on signal strength, enhancing performance.

Adaptive Networked Storage: For networked storage environments, RSS Scan aids in adjusting transmission parameters, contributing to network optimization.

**5. Differences Between N-Step Scan, F-Scan, and RSS Scan in Disk Scheduling:**

Nature of Scan:

N-Step Scan: Involves sequential scanning of disk sectors, similar to disk scheduling algorithms like FCFS (First-Come-First-Serve).

F-Scan: Focuses on analyzing frequency content, correlating with algorithms prioritizing specific data frequencies.

RSS Scan: Considers signal strength, aligning with algorithms optimizing based on the physical location of data on the disk.

Applications:

N-Step Scan: Optimal for scenarios with sequential data access patterns, reducing seek time.

F-Scan: Efficient for scenarios where certain frequencies or access patterns are critical for performance.

RSS Scan: Beneficial for scenarios involving wireless disk access or networked storage, optimizing based on signal strength.

Granularity:

N-Step Scan: Granularity determined by the number of steps (N) in the disk sectors.

F-Scan: Granularity determined by the frequency resolution of the data.

RSS Scan: Granularity influenced by the sensitivity of signal strength measurements.

6. Conclusion and Future Prospects in Disk Scheduling:

In conclusion, N-Step Scan, F-Scan, and RSS Scan prove valuable in optimizing disk access and retrieval based on distinct methodologies. Their applications in disk scheduling align with the need for efficient data management and reduced seek times.

**Future Prospects in Disk Scheduling:**

As technology evolves, future prospects in disk scheduling include:

Hybrid Approaches: Investigating the benefits of combining these scanning techniques within disk scheduling algorithms to achieve enhanced capabilities and adaptability.

Machine Learning Integration: Incorporating machine learning algorithms into disk scheduling for more intelligent prediction and optimization based on historical data access patterns.

Real-Time Disk Access: Advancements in real-time processing capabilities for disk scheduling, catering to dynamic data access requirements and minimizing latency.

Security in Disk Access: Exploring the potential of these scanning techniques in disk scheduling for security applications, such as anomaly detection and prevention of unauthorized access.

In conclusion, the analysis of N-Step Scan, F-Scan, and RSS Scan in the context of disk scheduling opens avenues for further research and innovation. Continued exploration of these methodologies will contribute to the development of more sophisticated and adaptive disk scheduling algorithms, addressing the evolving needs of data storage and retrieval systems.Project Report: Comparative Analysis of N-Step Scan, F-Scan, and RSS Scan in Disk Scheduling

**1. Introduction:**

In the domain of disk scheduling techniques, N-Step Scan, F-Scan, and RSS Scan represent three methodologies crucial for optimizing data access and retrieval. This project report aims to provide an in-depth understanding of these scanning techniques, explore their applications in the context of disk scheduling, and highlight the differences between them.

**2. N-Step Scan and Disk Scheduling:**

N-Step Scan in the context of disk scheduling involves dividing the disk into N sectors and sequentially scanning each sector. The granularity of the scan is determined by the number of steps (N). Applications of N-Step Scan in disk scheduling include:

Sequential Access: N-Step Scan ensures systematic access to data on the disk, minimizing seek time and improving data retrieval efficiency.

Optimization for Image Storage: In scenarios where images or files are stored sequentially, N-Step Scan aligns with the access pattern, reducing access times.

**3. F-Scan and Disk Scheduling:**

F-Scan, emphasizing frequency analysis, correlates with disk scheduling algorithms that prioritize specific data frequencies or access patterns. Applications of F-Scan in disk scheduling include:

Frequent Data Access: F-Scan algorithms optimize for frequently accessed data, reducing seek time and enhancing overall disk performance.

Audio or Video File Access: In scenarios where certain frequencies represent critical audio or video data, F-Scan can efficiently organize data access.

**4. RSS Scan and Disk Scheduling:**

RSS Scan, focused on signal strength, finds relevance in disk scheduling by considering the physical location of data on the disk. Applications of RSS Scan in disk scheduling include:

Optimizing Wireless Disk Access: In scenarios where disks are part of wireless storage systems, RSS Scan helps optimize access based on signal strength, enhancing performance.

Adaptive Networked Storage: For networked storage environments, RSS Scan aids in adjusting transmission parameters, contributing to network optimization.

**5. Differences Between N-Step Scan, F-Scan, and RSS Scan in Disk Scheduling:**

Nature of Scan:

N-Step Scan: Involves sequential scanning of disk sectors, similar to disk scheduling algorithms like FCFS (First-Come-First-Serve).

F-Scan: Focuses on analyzing frequency content, correlating with algorithms prioritizing specific data frequencies.

RSS Scan: Considers signal strength, aligning with algorithms optimizing based on the physical location of data on the disk.

**Applications:**

N-Step Scan: Optimal for scenarios with sequential data access patterns, reducing seek time.

F-Scan: Efficient for scenarios where certain frequencies or access patterns are critical for performance.

RSS Scan: Beneficial for scenarios involving wireless disk access or networked storage, optimizing based on signal strength.

**Granularity:**

N-Step Scan: Granularity determined by the number of steps (N) in the disk sectors.

F-Scan: Granularity determined by the frequency resolution of the data.

RSS Scan: Granularity influenced by the sensitivity of signal strength measurements.

**6. Conclusion and Future Prospects in Disk Scheduling:**

In conclusion, N-Step Scan, F-Scan, and RSS Scan prove valuable in optimizing disk access and retrieval based on distinct methodologies. Their applications in disk scheduling align with the need for efficient data management and reduced seek times.

**Future Prospects in Disk Scheduling:**

As technology evolves, future prospects in disk scheduling include:

Hybrid Approaches: Investigating the benefits of combining these scanning techniques within disk scheduling algorithms to achieve enhanced capabilities and adaptability.

Machine Learning Integration: Incorporating machine learning algorithms into disk scheduling for more intelligent prediction and optimization based on historical data access patterns.

Real-Time Disk Access: Advancements in real-time processing capabilities for disk scheduling, catering to dynamic data access requirements and minimizing latency.

Security in Disk Access: Exploring the potential of these scanning techniques in disk scheduling for security applications, such as anomaly detection and prevention of unauthorized access.

In conclusion, the analysis of N-Step Scan, F-Scan, and RSS Scan in the context of disk scheduling opens avenues for further research and innovation. Continued exploration of these methodologies will contribute to the development of more sophisticated and adaptive disk scheduling algorithms, addressing the evolving needs of data storage and retrieval systems.

**1. Lidar Scanning and Disk Scheduling:**

Lidar scanning, primarily used for creating 3D maps, shares a parallel with disk scheduling in terms of its precision and systematic coverage. In disk scheduling, algorithms like SCAN (Elevator) and C-SCAN move the disk arm in a systematic manner, resembling Lidar's ability to sweep across an environment. The meticulous mapping and navigation Lidar provides can be compared to the efficient data access achieved by SCAN and C-SCAN in disk scheduling, ensuring systematic retrieval or storage of data.

**2. Barcode Scanning and Disk Scheduling:**

Barcode scanning is akin to the concept of scanning in disk scheduling algorithms. Just as a barcode scanner quickly reads and processes information from a barcode, disk scheduling algorithms like Shortest Seek Time First (SSTF) aim to minimize seek time by rapidly accessing the closest data. Both technologies prioritize efficiency and speed in information retrieval, essential for industries where swift data access is critical, such as retail or logistics.

**3. CT Scanning and Disk Scheduling:**

CT scanning involves capturing cross-sectional images layer by layer, reminiscent of how disk scheduling algorithms like C-LOOK operate. These algorithms focus on a specific range of tracks, moving the disk arm back and forth within that range. Similarly, CT scanning provides detailed images of specific slices of the body. Both emphasize a targeted and systematic approach, avoiding unnecessary traversal.

**4. Document Scanning and Disk Scheduling:**

Document scanning aligns with disk scheduling principles concerning the efficient organization and retrieval of data. In disk scheduling, algorithms like First-Come-First-Serve (FCFS) and LOOK ensure a sequential approach to accessing data. Similarly, document scanning transforms physical documents into a structured digital format, allowing for organized and sequential data retrieval.

**5. N-Step Scan, F-Scan, and RSS Scan in Disk Scheduling:**

N-Step Scan, F-Scan, and RSS Scan can be related to the diverse disk scheduling algorithms available:

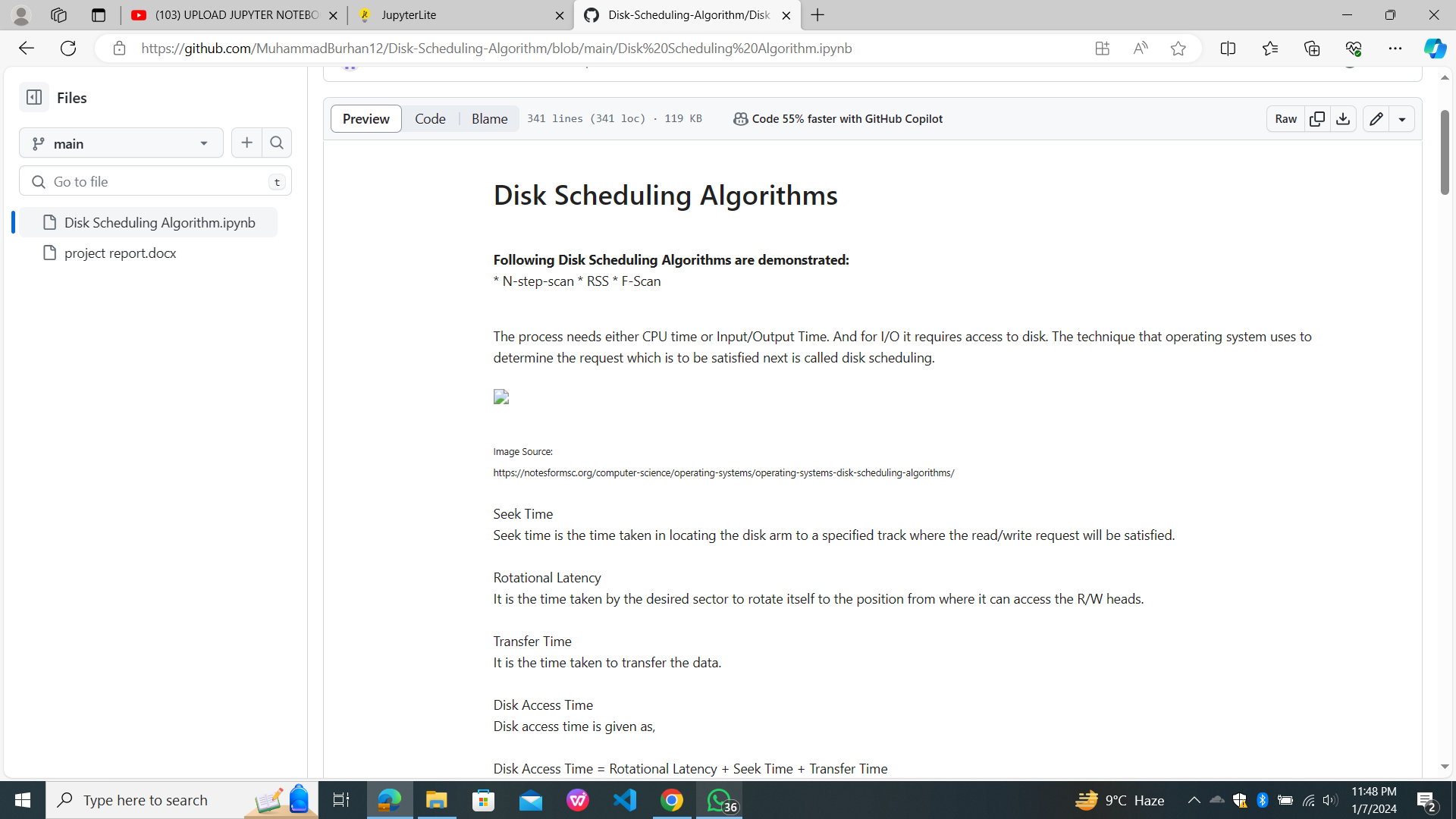
N-Step Scan relates to algorithms like SCAN or C-SCAN, where the disk arm moves in steps across the disk surface, systematically accessing data.

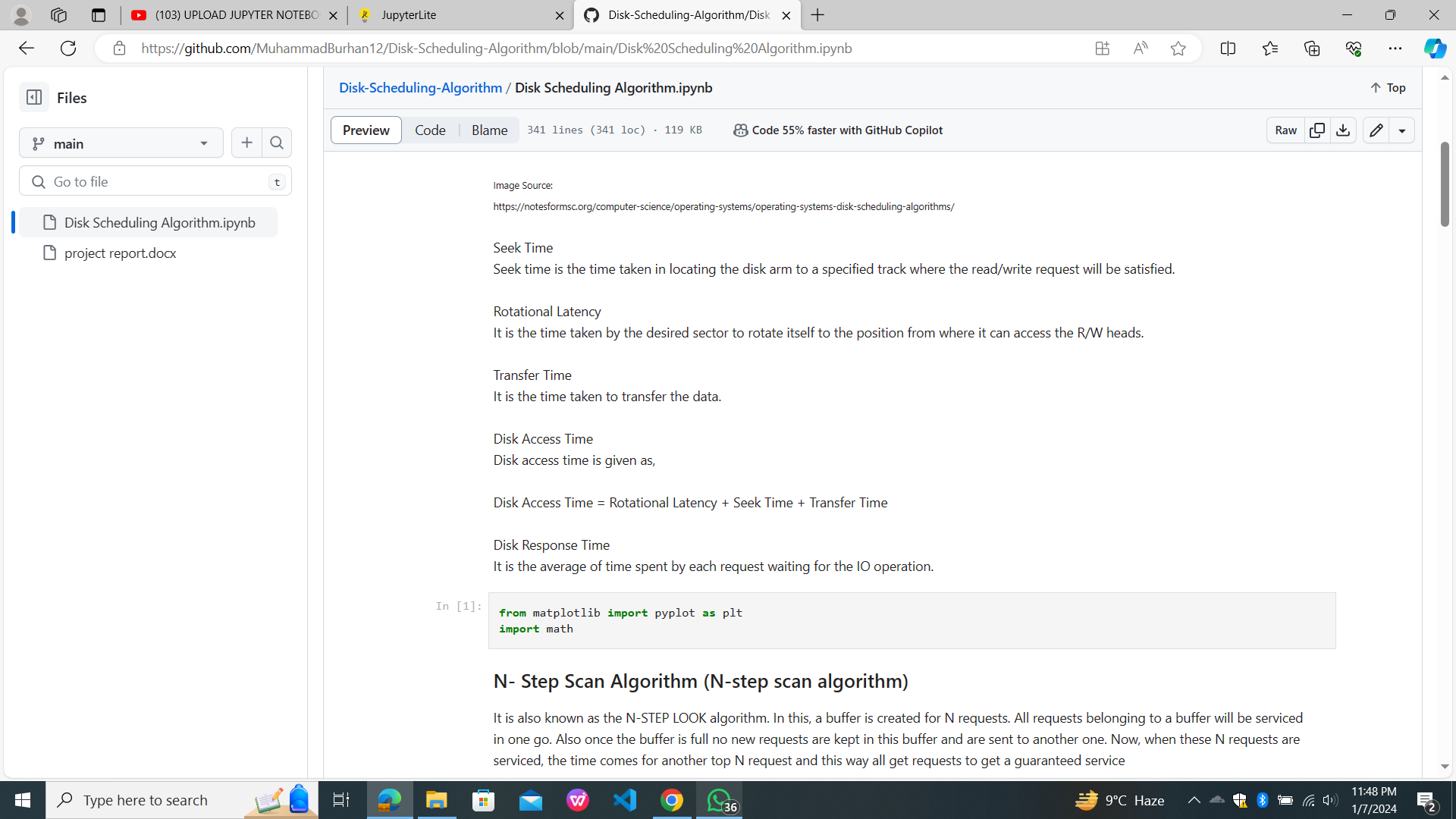
F-Scan, emphasizing frequency analysis, can be associated with algorithms that prioritize frequently accessed data or sectors, optimizing for data with higher demand.

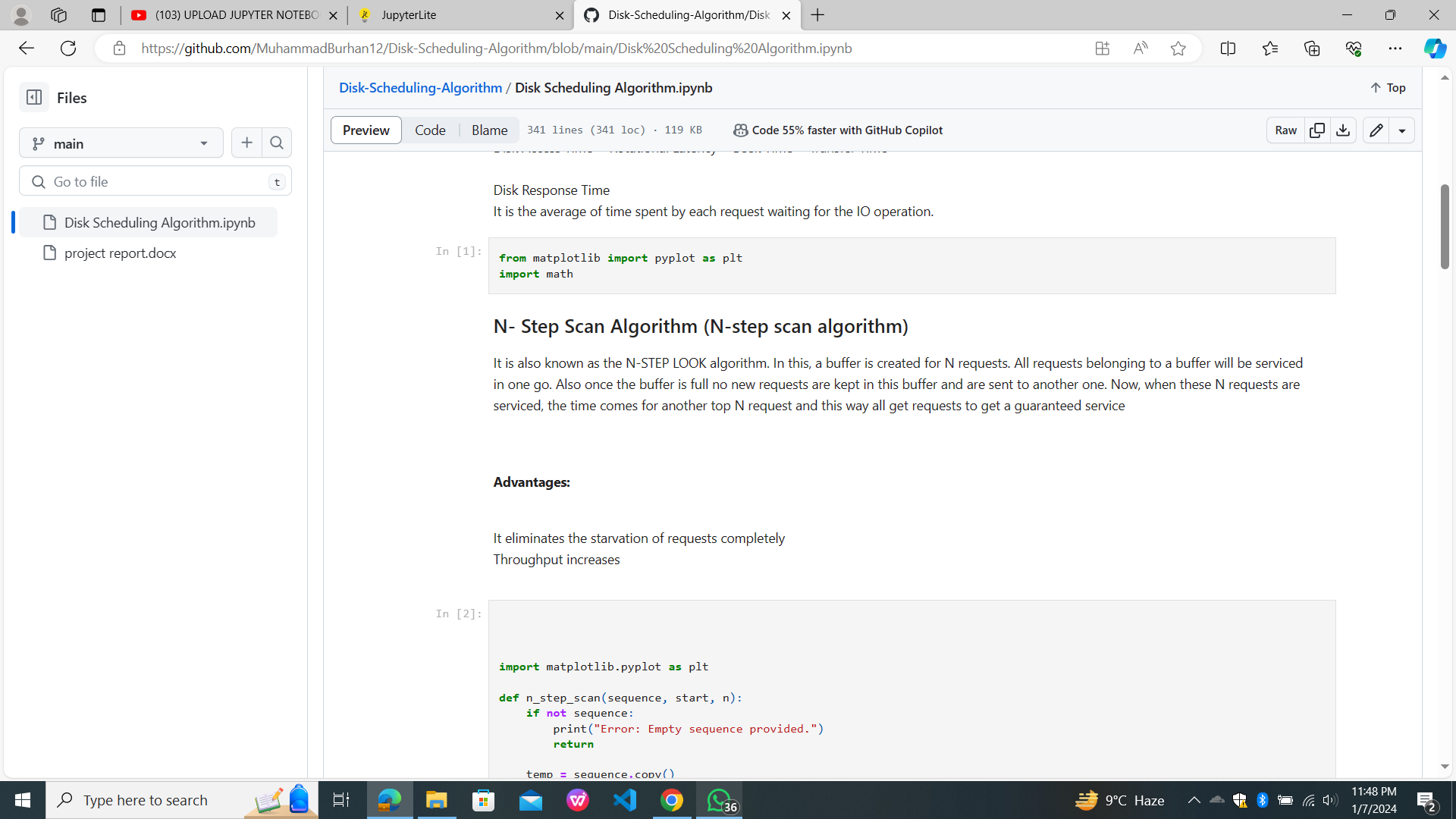
RSS Scan, which focuses on signal strength, can be likened to algorithms that consider the physical location of data on the disk, aiming to reduce seek time and enhance performance.

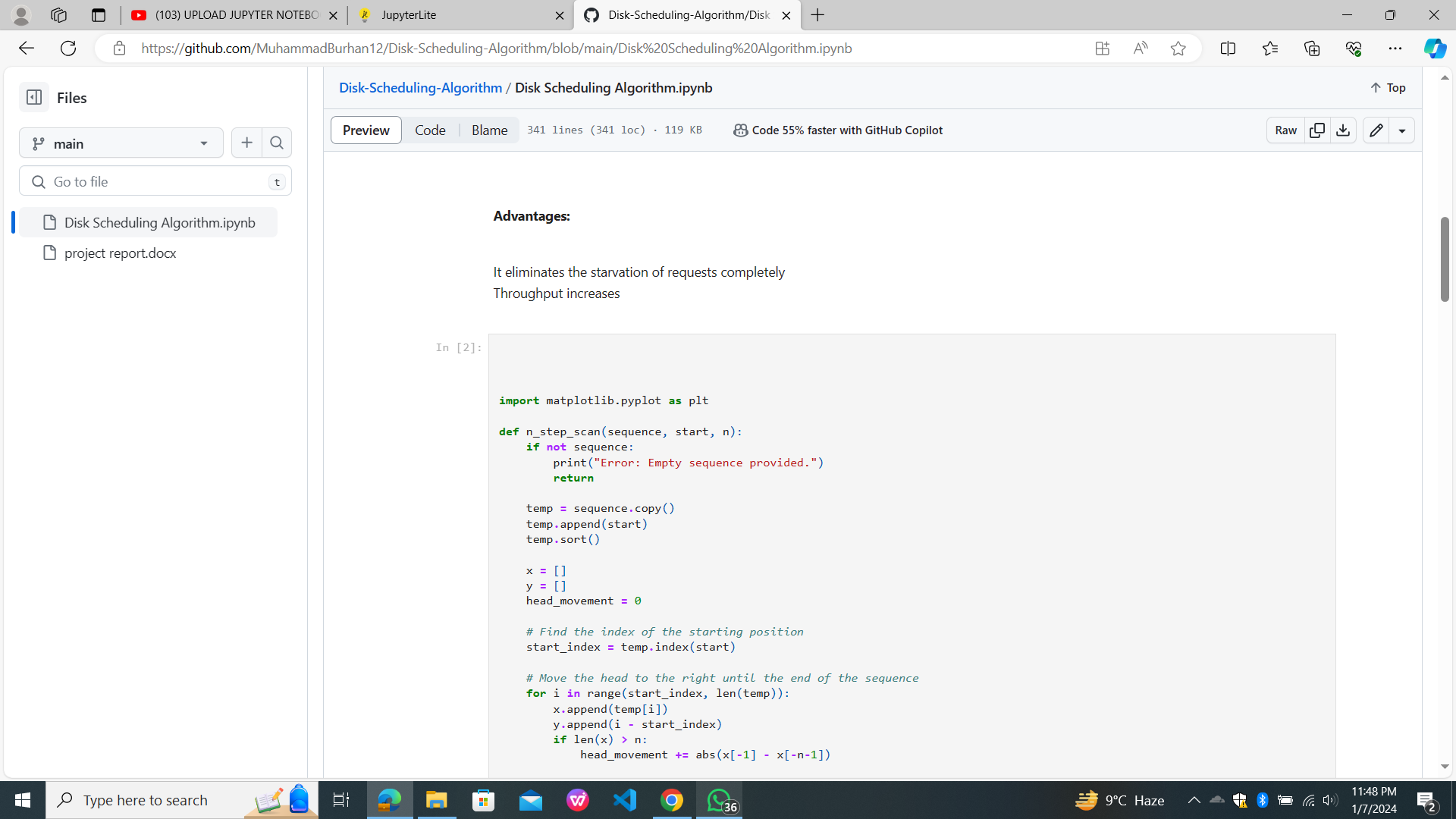
Understanding the parallels between these scanning types and disk scheduling provides insights into how diverse scanning methodologies find relevance in optimizing data access and retrieval from disk storage systems.

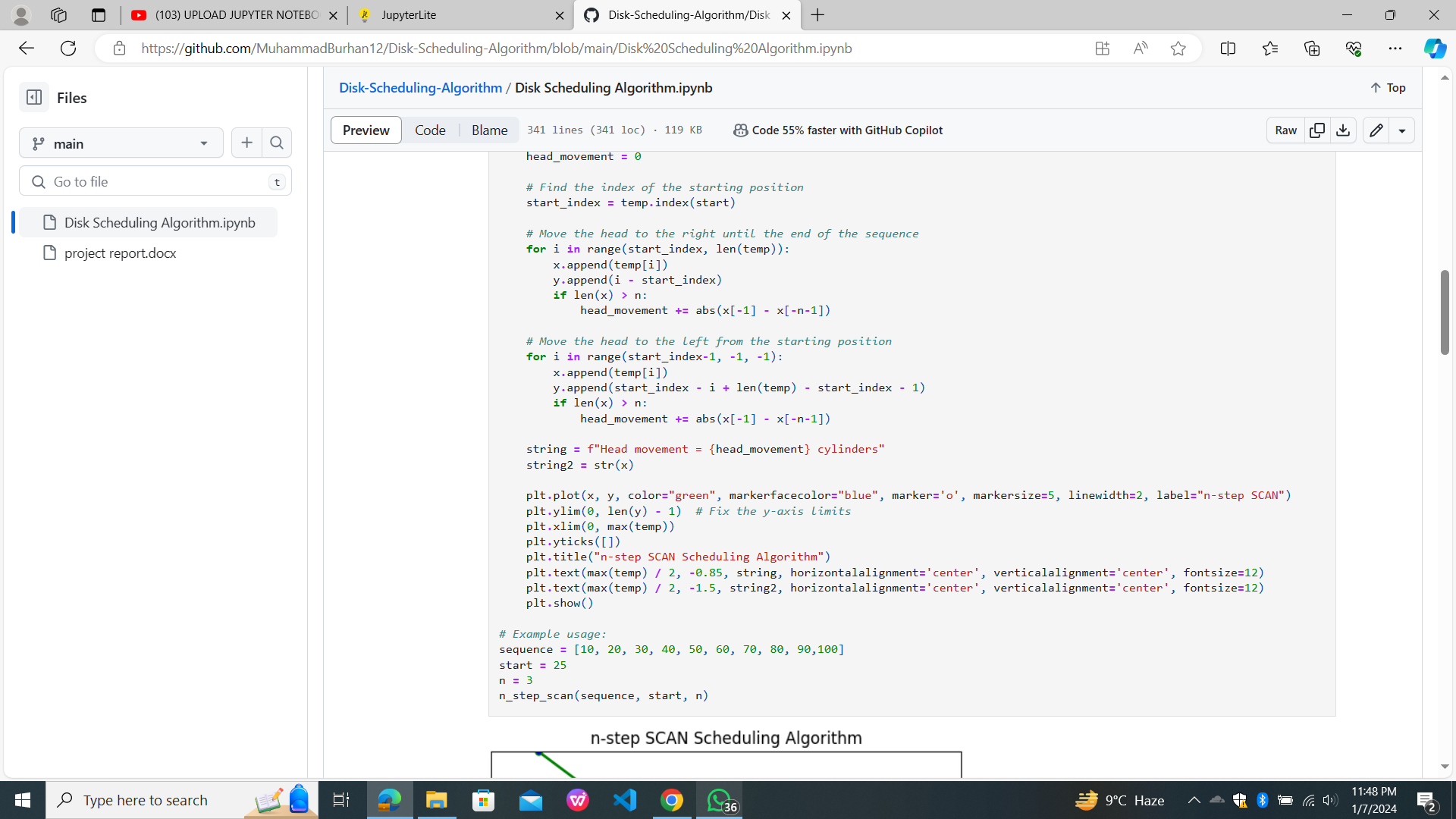
Code and its output:

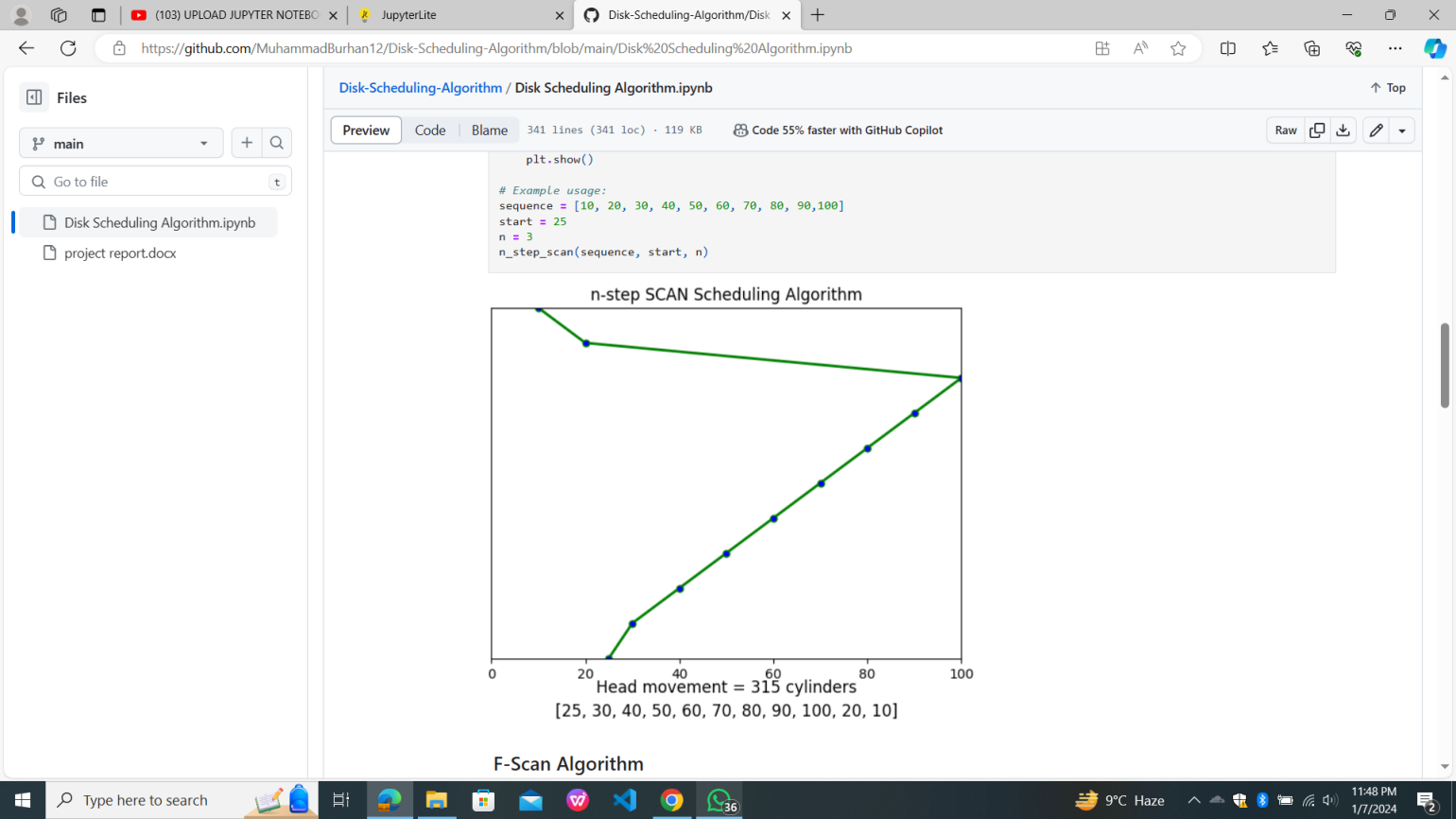


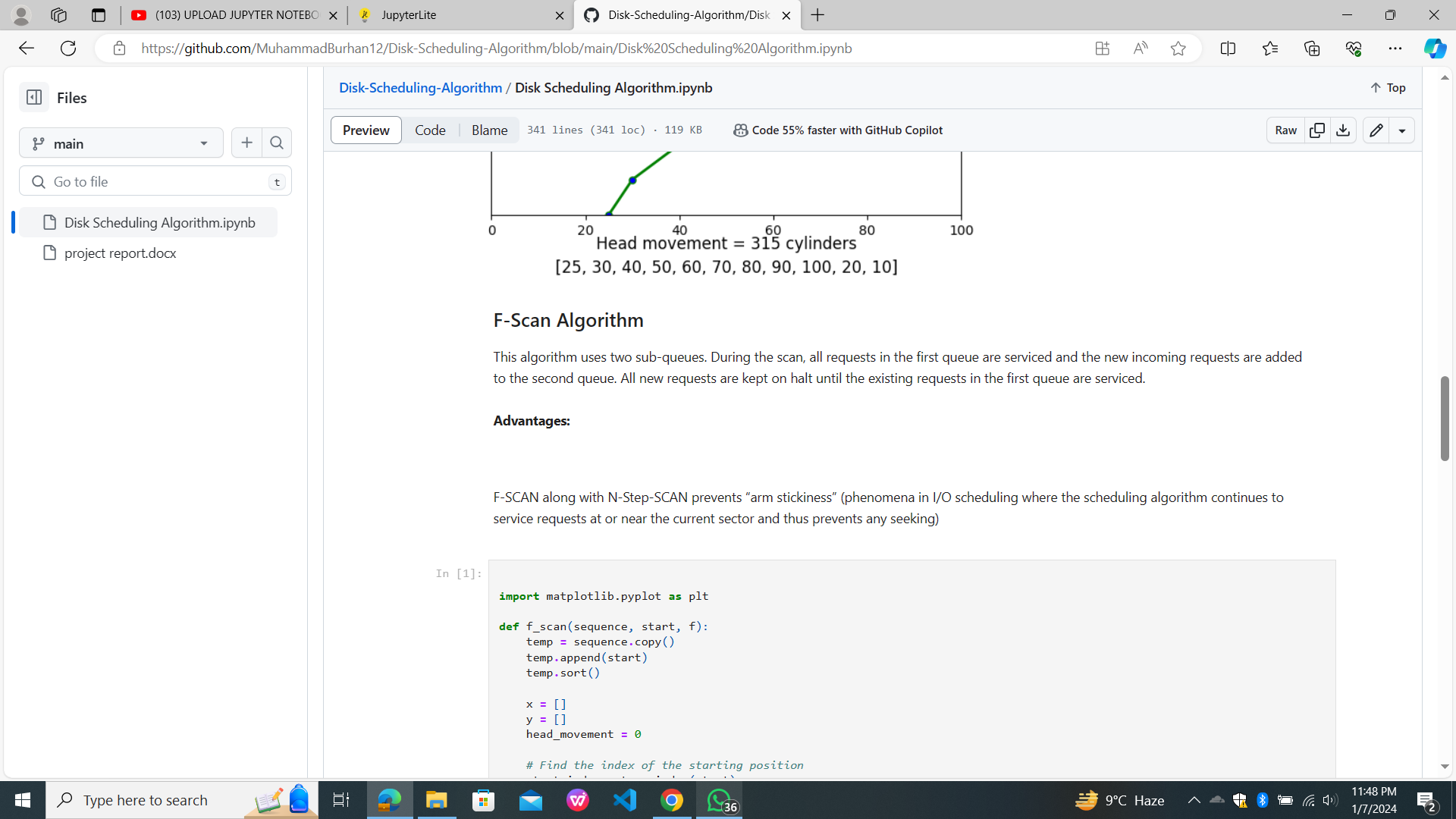


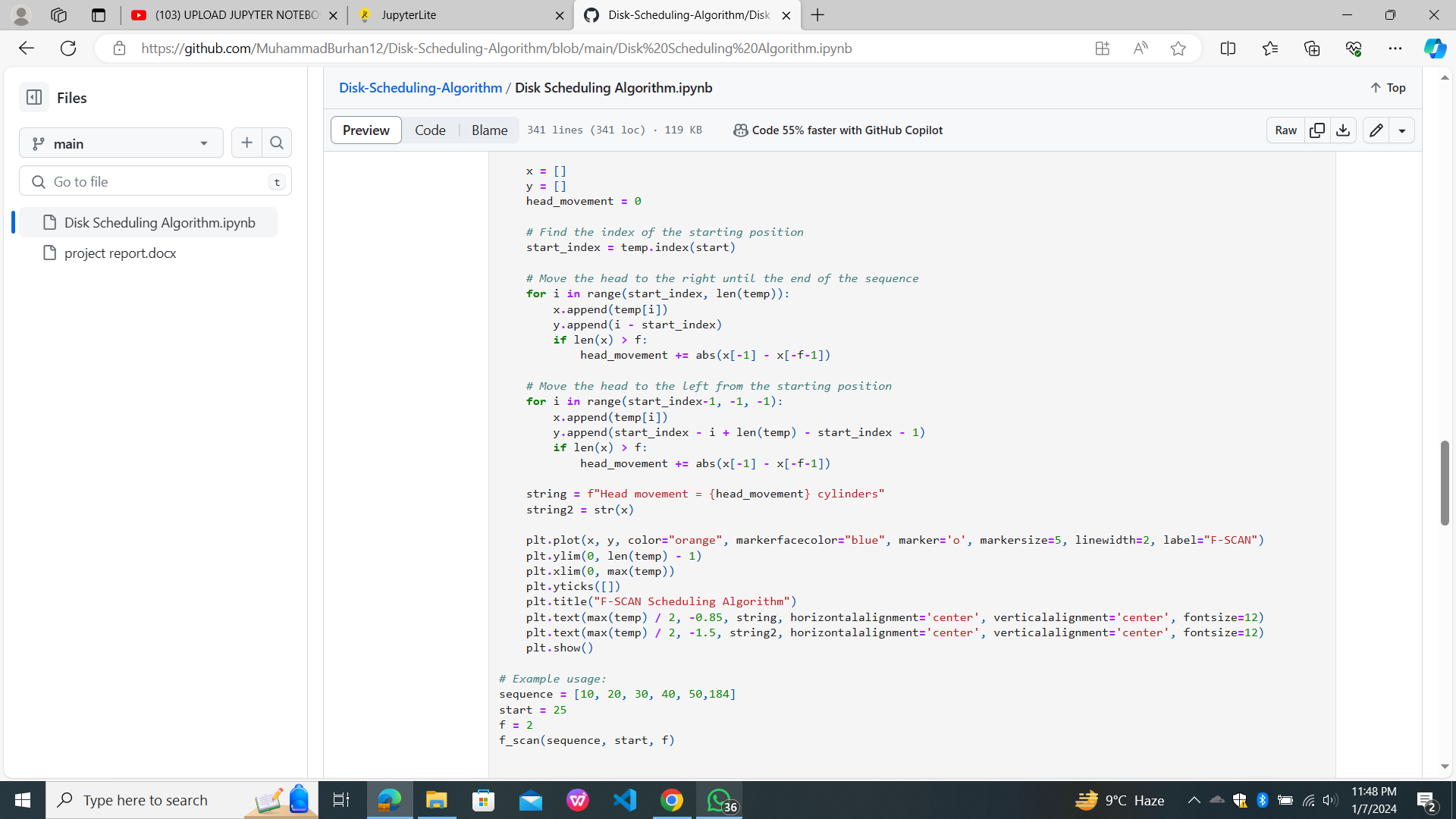


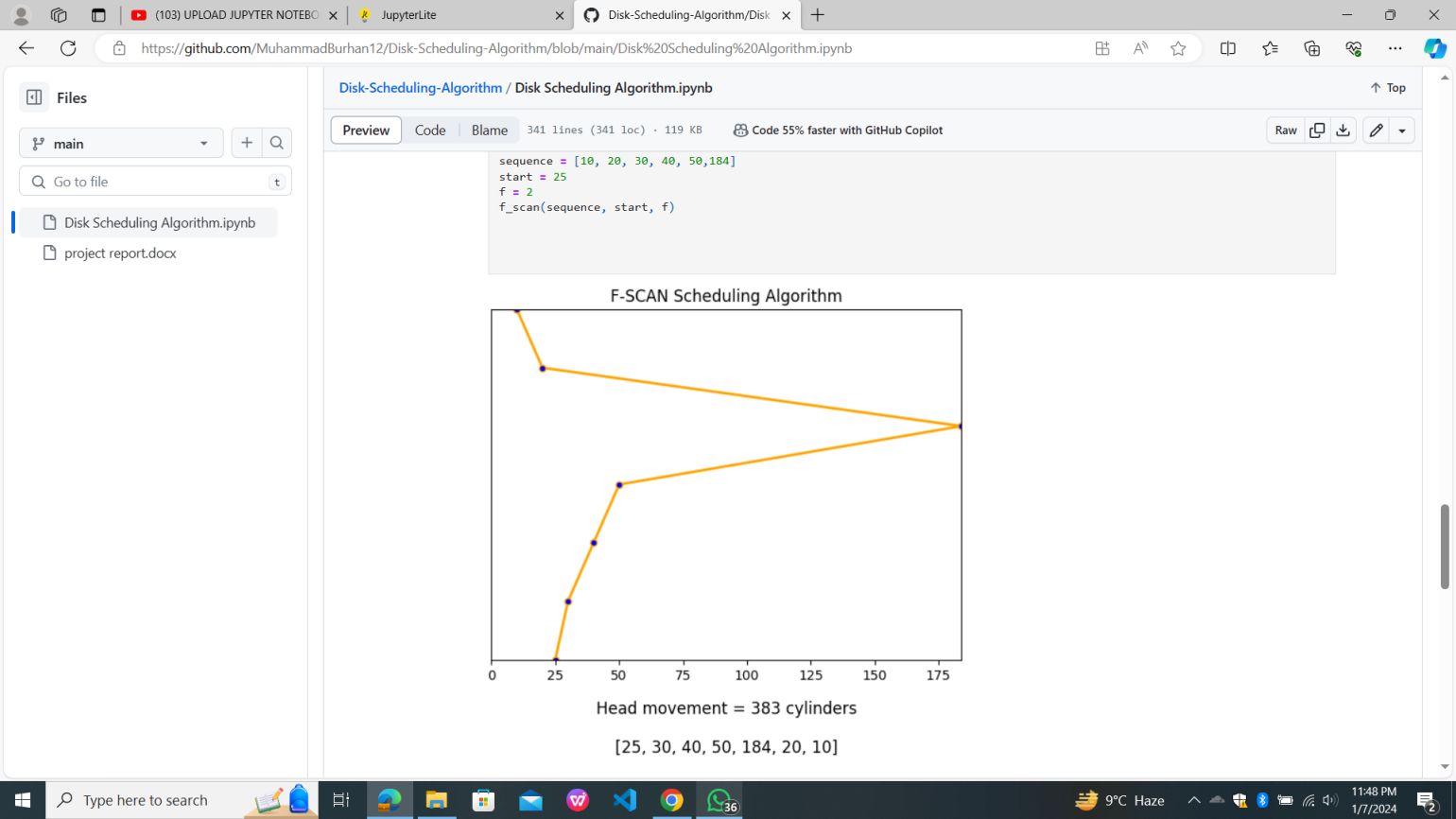


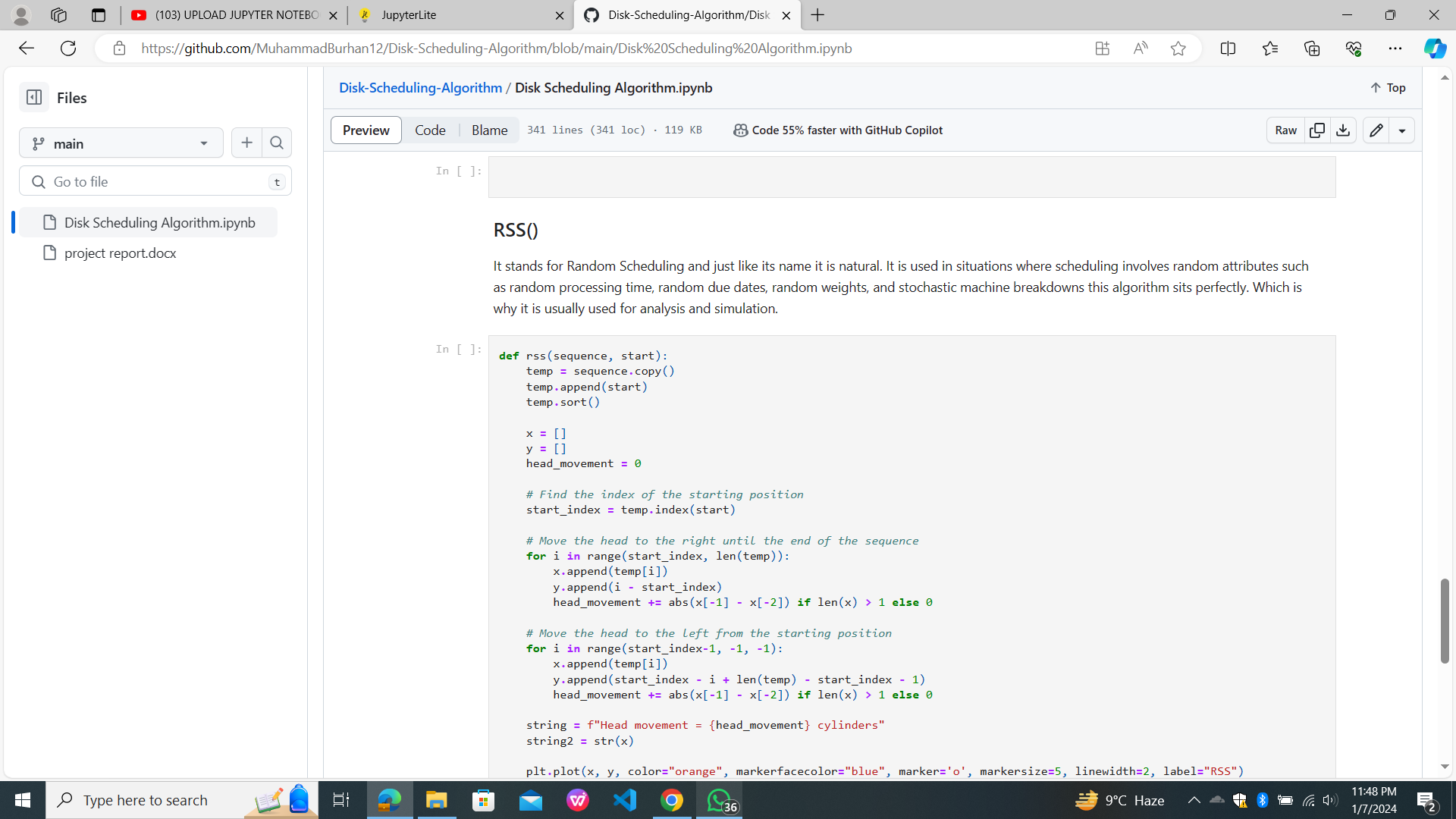


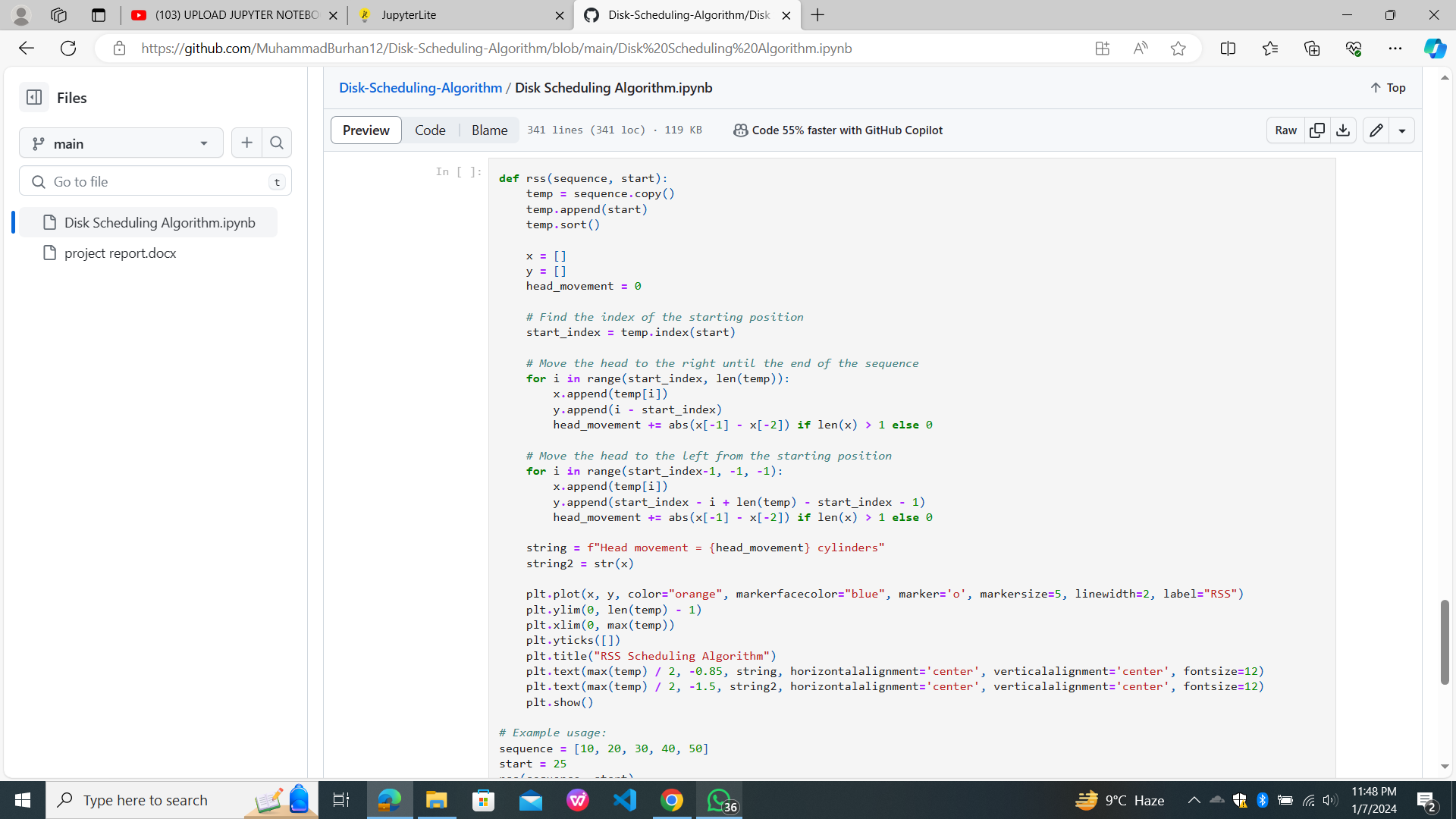


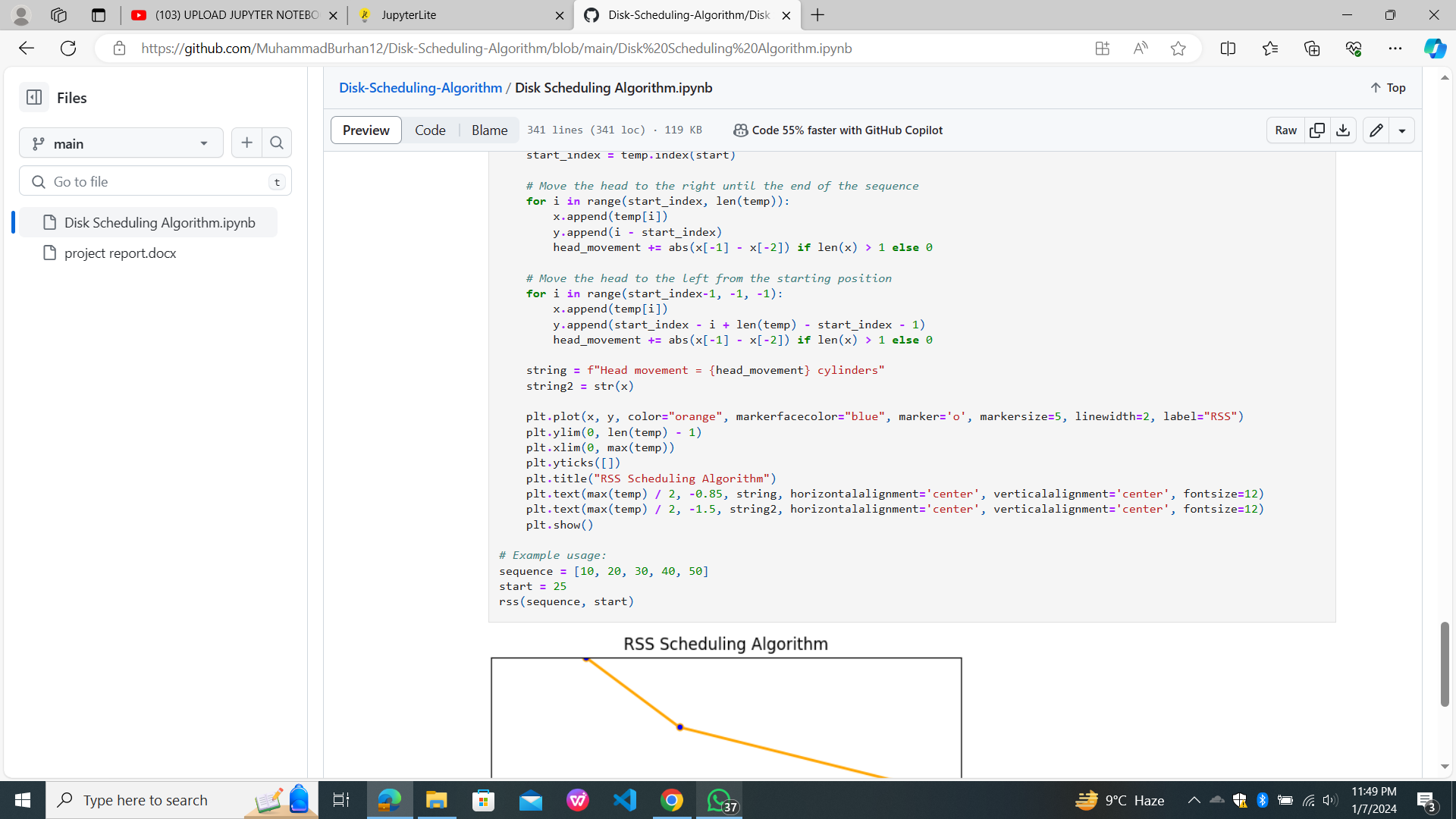


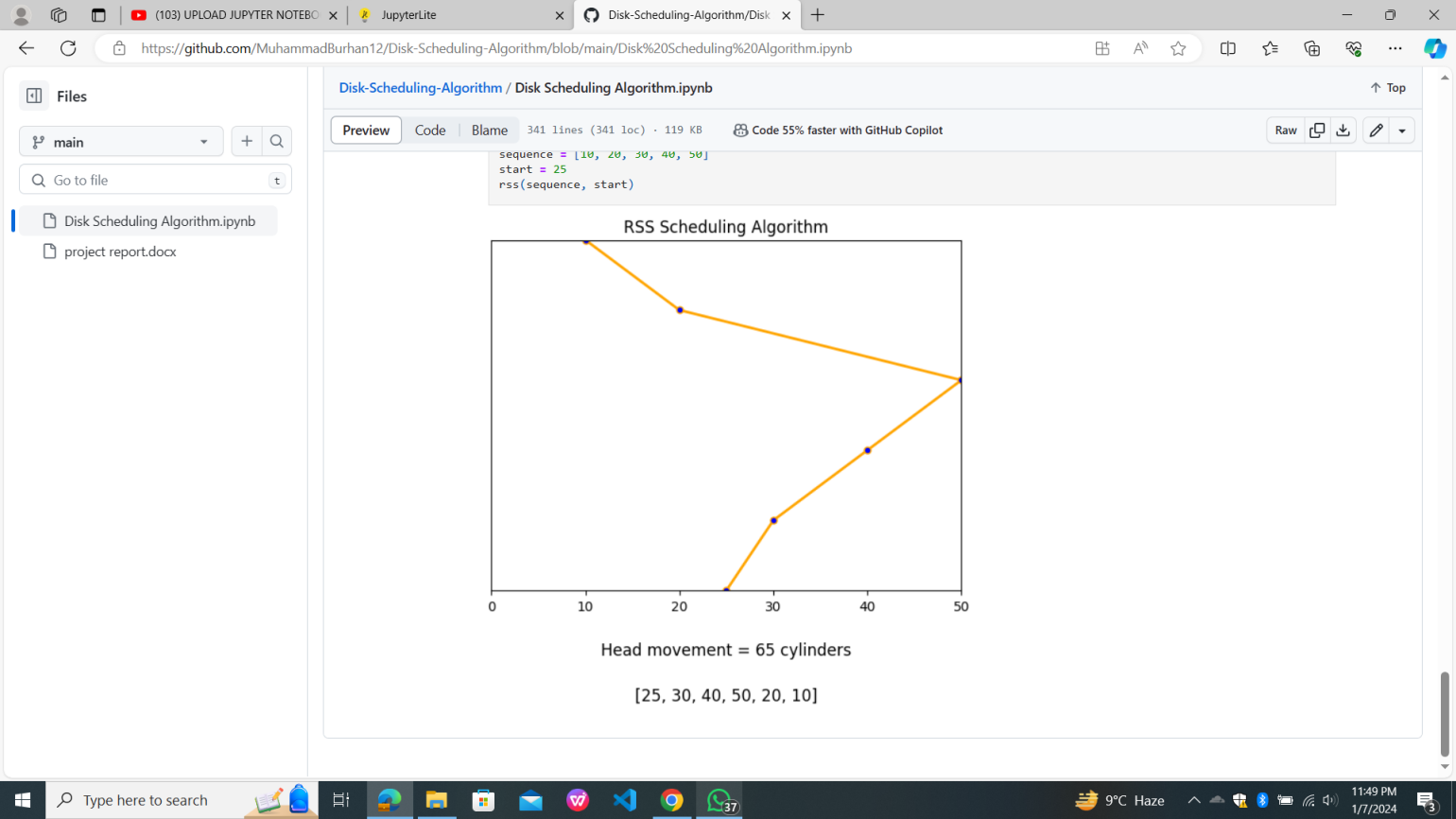












READ ME File :

# Disk Scheduling Algorithms

**# Disk Scheduling Algorithms in Python**

<img src="https://img.shields.io/badge/Operating Systems-Disk Scheduling Algorithms-<green>"> <img src="https://img.shields.io/static/v1?label=Language&message=Python,Jupyter&color=blue">

<br>

Implemented various Disk Scheduling Algorithms in Python

<br>

<br><b> Following Disk Scheduling Algorithms are demonstrated:<br></b>

    1. N-Step Scan<br>

    2. F-scan<br>

    3. RSS<br>

<br>

The process needs either CPU time or Input/Output Time. And for I/O it requires access to disk. The technique that operating system uses to determine the request which is to be satisfied next is called disk scheduling.<br>

<br>

<img src="image.png">

<font size = 1>

<br>Image Source:<br>

https://notesformsc.org/computer-science/operating-systems/operating-systems-disk-scheduling-algorithms/<br> </font>

<br>

Seek Time<br>

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

<br><br>

Rotational Latency<br>

It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.

<br><br>

Transfer Time<br>

It is the time taken to transfer the data.

<br><br>

Disk Access Time<br>

Disk access time is given as,

<br><br>

Disk Access Time = Rotational Latency + Seek Time + Transfer Time

<br><br>

Disk Response Time<br>

It is the average of time spent by each request waiting for the IO operation.

RSS Scheduling Algorithm

Description

RSS (Rotational Shortest Seek) is a disk scheduling algorithm that aims to minimize the rotational latency by selecting the disk request with the shortest seek time. It considers both the distance to the requested track and the rotational position of the disk.

Usage:

from disk\_scheduling import rss

**# Example usage:**

sequence = [10, 20, 30, 40, 50]

start = 25

rss(sequence, start)

n-step SCAN Scheduling Algorithm

Description

n-step SCAN is an extension of the SCAN algorithm. It divides the disk into n zones and services requests in a round-robin manner within each zone. It helps prevent starvation of requests in the outer tracks.

Usage:

from disk\_scheduling import n\_step\_scan

**# Example usage:**

sequence = [10, 20, 30, 40, 50]

start = 25

n = 3

n\_step\_scan(sequence, start, n)

F-scan disk scheduling algorithm:

F-SCAN is a disk scheduling algorithm designed to optimize head movement during disk I/O operations. It utilizes a fixed-size queue to keep track of recently accessed disk locations, scanning the disk in both forward and backward directions.

**## F-SCAN Scheduling Algorithm**

**### Description**

F-SCAN is a disk scheduling algorithm that uses a fixed-size queue to keep track of the recently accessed disk locations. It scans the disk in a forward and backward manner, minimizing head movement.

**## How F-SCAN Works**

1. **\*\*Initialization:\*\*** The algorithm starts with a given sequence of disk requests and the initial head position.

2. **\*\*Sorting:\*\*** The requested disk locations, along with the initial head position, are sorted in ascending order.

3. **\*\*Forward Scan:\*\*** The head moves to the right, serving disk requests in the sorted order until the end of the sequence. The algorithm records the disk locations and the distance covered by the head.

4. **\*\*Backward Scan:\*\*** The head then moves back to the starting position and continues to the left, serving disk requests in reverse order. Again, the algorithm records the disk locations and the distance covered by the head.

5. **\*\*Head Movement Calculation:\*\*** The total head movement is calculated as the sum of the absolute differences between consecutive disk locations during both forward and backward scans.

**## Usage**

To use the F-SCAN algorithm in your Python program:

```python

from disk\_scheduling import f\_scan

# Example usage:

sequence = [10, 20, 30, 40, 50]

start = 25

f = 2

f\_scan(sequence, start, f)

Example Visualization

The algorithm's execution can be visualized through a plot, showing the movement of the disk head and providing information about the total head movement.