Practical No 10

Aim : Disk Scheduling and Simple File System Designs

10.1: Simulate FCFS,SSTF,C-SCAN,C-LOOK,RSS for disk head movement
10.2:Design a basic file system structure with block allocation, directory management, and file operations (create, read, delete)

```
import random

class DiskScheduling:
    def __init__(self, requests, head, diskSize=200):
        """
        :param requests: list of disk track requests
        :param head: initial head position
        :param diskSize: total size of disk (default 200)
        """
        self.requests = requests[:]  # make a copy
        self.head = head
        self.diskSize = diskSize

    def fcfs(self):
        """First Come First Serve"""
        distance, head = 0, self.head
        order = []
        for req in self.requests:
            distance += abs(head - req)
            order.append(req)
            head = req
        return order, distance

    def sstf(self):
        """Shortest Seek Time First"""
        distance, head = 0, self.head
        requests = self.requests[:]
        order = []
        while requests:
            closest = min(requests, key=lambda x: abs(x - head))
            distance += abs(head - closest)
            order.append(closest)
            head = closest
            requests.remove(closest)
        return order, distance

    def cscan(self):
        """Circular SCAN"""
        distance, head = 0, self.head
        requests = sorted(self.requests)
        order = []

        right = [r for r in requests if r >= head]
        left = [r for r in requests if r < head]
        # Go rightwards till the end
        for r in right:
            distance += abs(head - r)
            order.append(r)
            head = r

        # Jump to the beginning and service left
        if left:
            distance += abs(self.diskSize - 1 - head)  # go to end
            distance += self.diskSize - 1              # jump to beginning
            head = 0
            for r in left:
                distance += abs(head - r)
                order.append(r)
                head = r

        return order, distance

    def clook(self):
        """Circular LOOK"""
        distance, head = 0, self.head
        requests = sorted(self.requests)
        order = []

        right = [r for r in requests if r >= head]
        left = [r for r in requests if r < head]

        # Service right side
        for r in right:
            distance += abs(head - r)
            order.append(r)
            head = r

        # Jump to smallest request on the left
        if left:
            distance += abs(head - left[0])
            head = left[0]
            for r in left:
                distance += abs(head - r)
                order.append(r)
                head = r

        return order, distance

    def rss(self):
        """Random Scheduling"""
        distance, head = 0, self.head
        requests = self.requests[:]
        order = []
        random.shuffle(requests)
        for r in requests:
            distance += abs(head - r)
            order.append(r)
            head = r
        return order, distance

# Example Usage
if __name__ == "__main__":
    requests = [82, 170, 43, 140, 24, 16, 190]
    head = 50
    diskSize = 200

    algo = DiskScheduling(requests, head, diskSize)

    fcfsOrder, fcfsDistance = algo.fcfs()
    print(f"FCFS order: {fcfsOrder}")
    print(f"FCFS total distance: {fcfsDistance}")

    sstfOrder, sstfDistance = algo.sstf()
    print(f"SSTF order: {sstfOrder}")
    print(f"SSTF total distance: {sstfDistance}")

    clookOrder, clookDistance = algo.clook()
    print(f"C-LOOK order: {clookOrder}")
    print(f"C-LOOK total distance: {clookDistance}")

    cscanOrder, cscanDistance = algo.cscan()
    print(f"C-SCAN order: {cscanOrder}")
    print(f"C-SCAN total distance: {cscanDistance}")

    rssOrder, rssDistance = algo.rss()
    print(f"RSS order: {rssOrder}")
    print(f"RSS total distance: {rssDistance}")
```

```
FCFS order: [82, 170, 43, 140, 24, 16, 190]
FCFS total distance: 642
SSTF order: [43, 24, 16, 82, 140, 170, 190]
SSTF total distance: 208
C-LOOK order: [82, 140, 170, 190, 16, 24, 43]
C-LOOK total distance: 341
C-SCAN order: [82, 140, 170, 190, 16, 24, 43]
C-SCAN total distance: 391
RSS order: [16, 24, 190, 82, 140, 170, 43]
RSS total distance: 531
```