OS - A7 REPORT

CS20B008

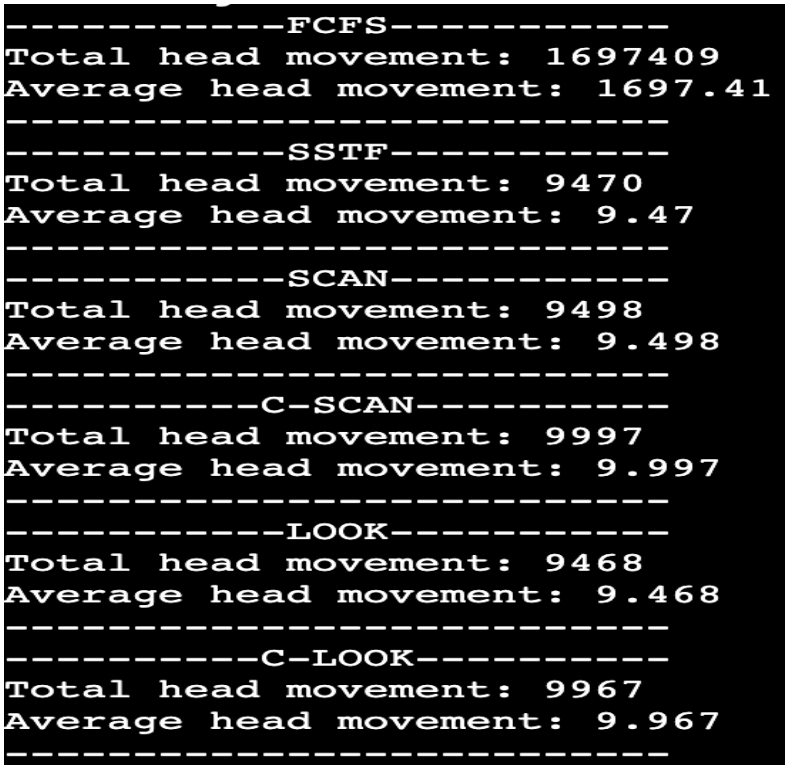
- Overview of disk Scheduling algorithms
    1. **FCFS**: In FCFS, the requests are addressed in the order they arrive in the disk queue.
    2. **SSTF**: In SSTF (Shortest Seek Time First), requests having the shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first.
    3. **SCAN**: In SCAN algorithm the disk arm moves in a particular direction and services the requests coming in its path and after reaching the end of the disk, it reverses its direction and again services the request arriving in its path.
    4. **C-SCAN**: In C-SCAN algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there.
    5. **LOOK**: It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only.
    6. **C-LOOK**: As LOOK is similar to SCAN algorithm, in a similar way, C-LOOK is similar to C-SCAN disk scheduling algorithm. In C-LOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request.

- Comparison of different algorithms

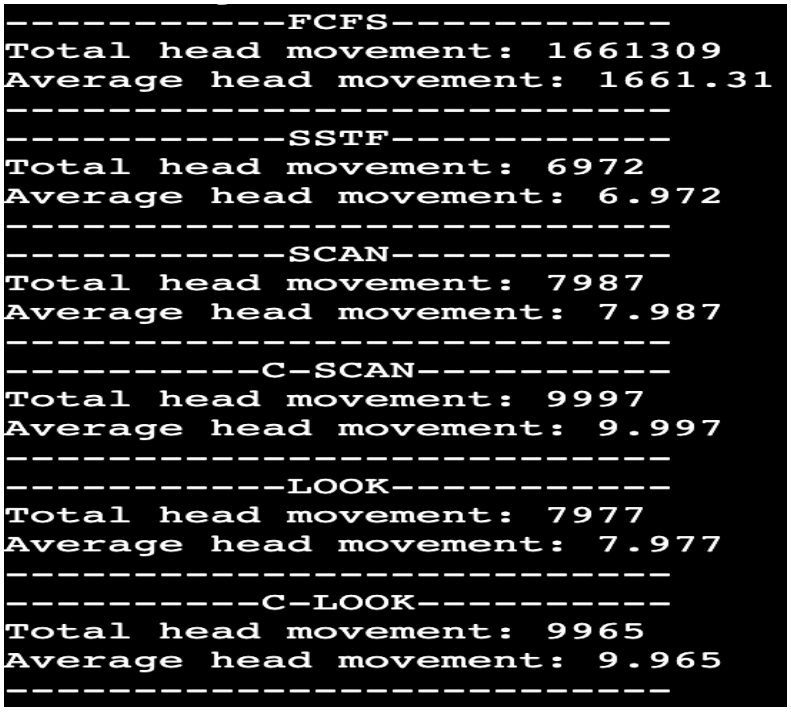
  We are servicing a disk with 5,000 cylinders numbered 0 to 4,999.
  We are generating a random series of 1,000-cylinder requests and servicing them according to each of the algorithms listed above.
  We then generate the average & total amount of head movement for the requests for each algorithm.

Position of initial disk head:

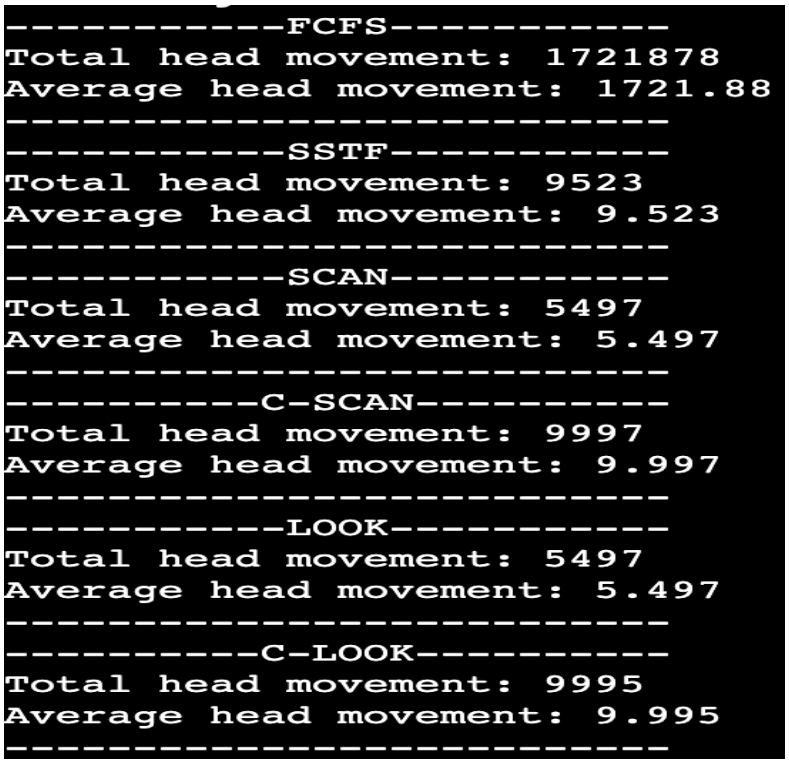
1. 500



2. 2000



3. **4500**



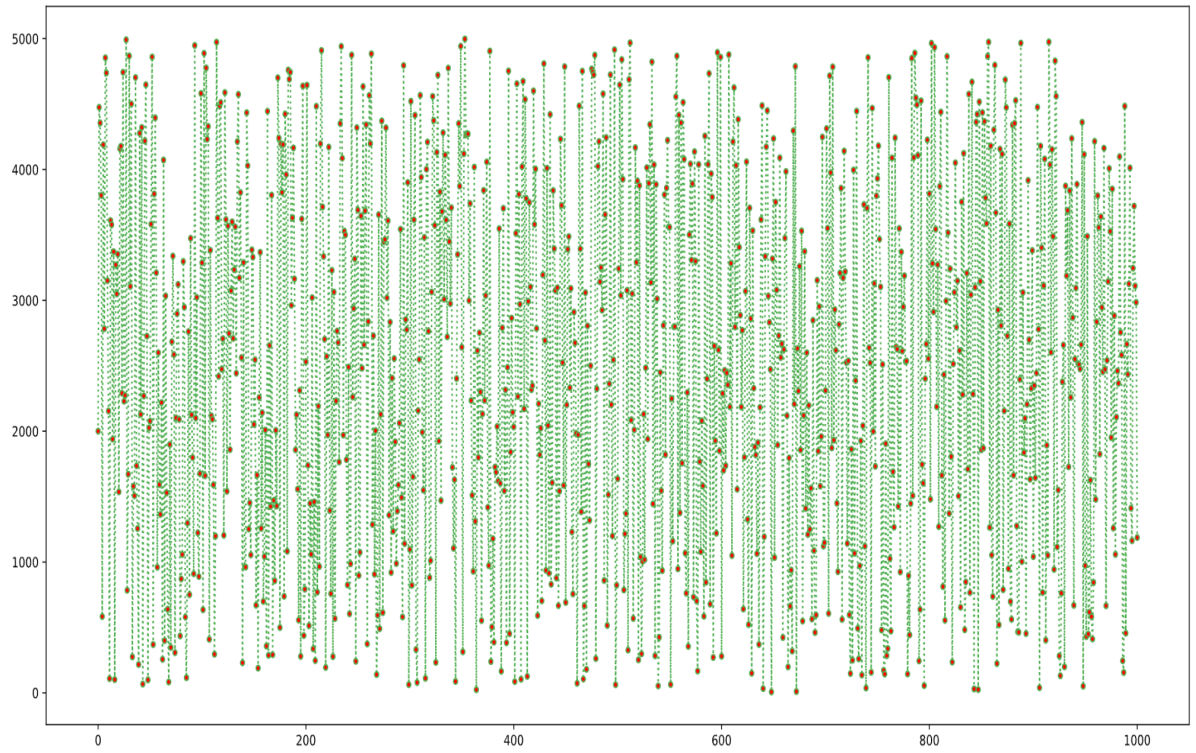
Inference:

- FCFS algorithm has the largest disk head movement, which is quite obvious because it's like a brute force algorithm.
- Circular algorithms (C-LOOK AND C-SCAN) have higher head movement than SCAN and LOOK but much less than FCFS.
- SCAN and LOOK have the approximately the same head movement, but SCAN will always have higher head movement than LOOK, this can be proved mathematically.
- C-SCAN and C-LOOK have the approximately the same head movement, but C-SCAN will always have higher head movement than C-LOOK, this can be proved mathematically.
- SSTF algorithm sometimes has higher head movement that SCAN and LOOK and sometimes lower than them.
- We infer that the best algorithm is: **SSTF** or **SCAN**.

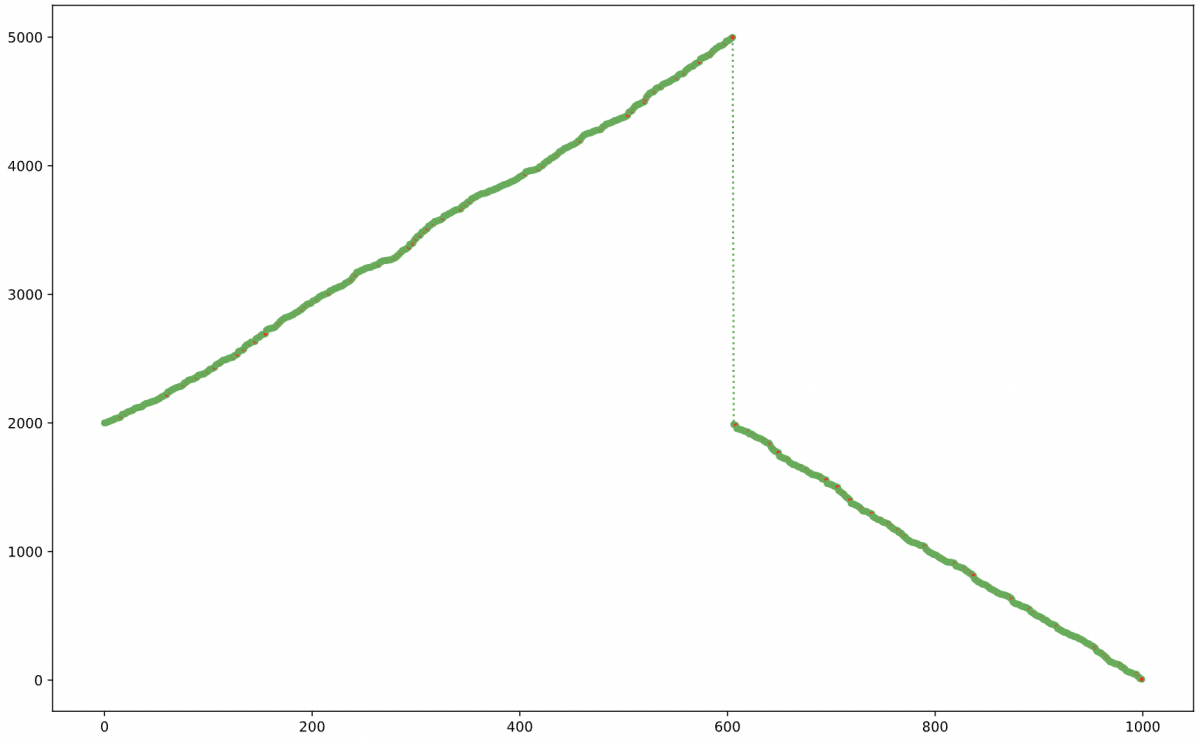
- ## Graphical representation of the algorithms

  These graphs were generated using the python library **matplotlib**.
  The initial disk head is at 2000.
  There are a total of 1000 requests(randomly generated).

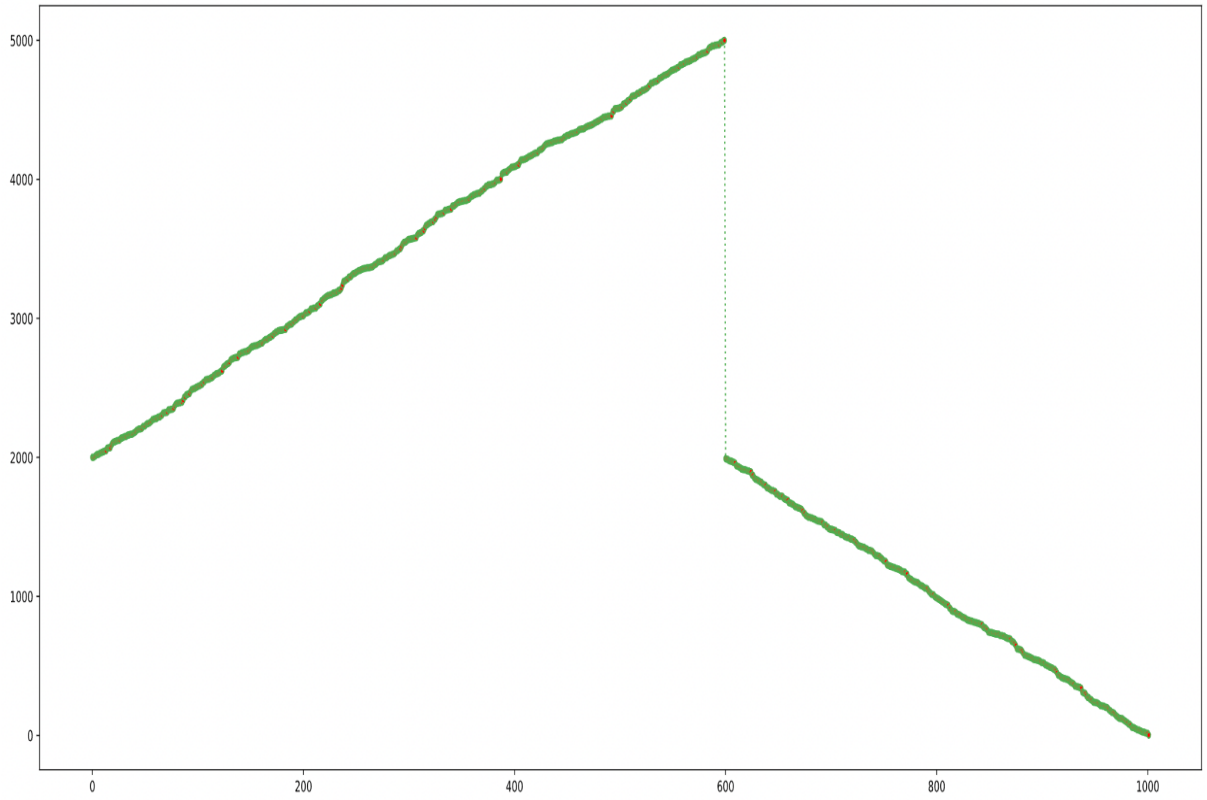
1. FCFS



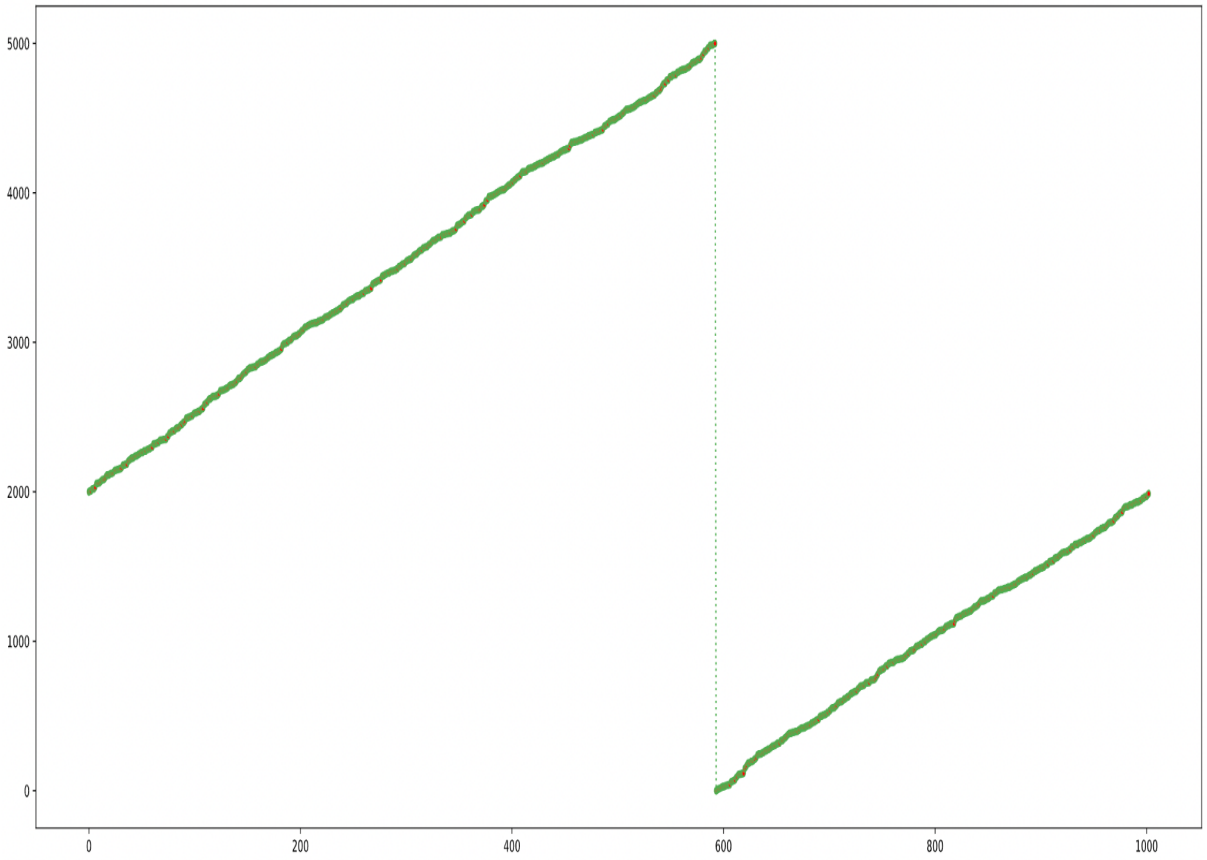
2. SSTF



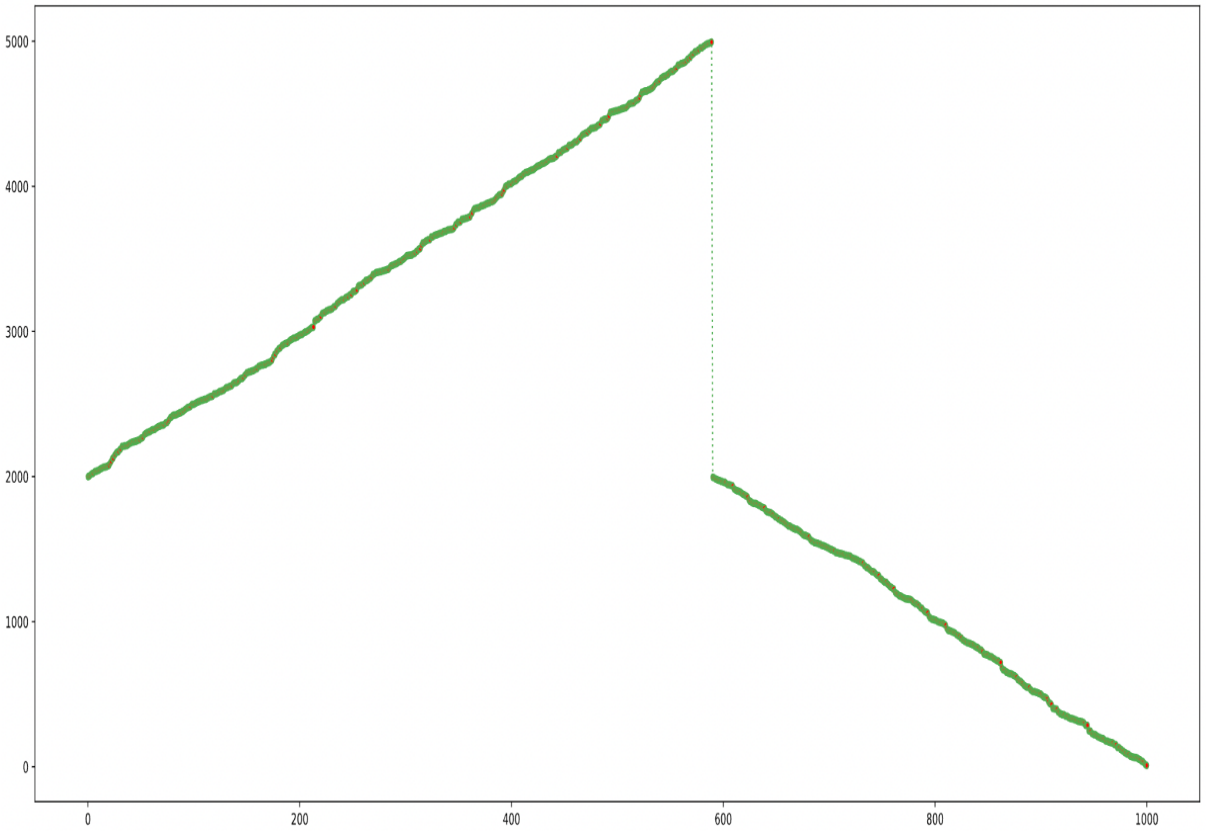
3. SCAN



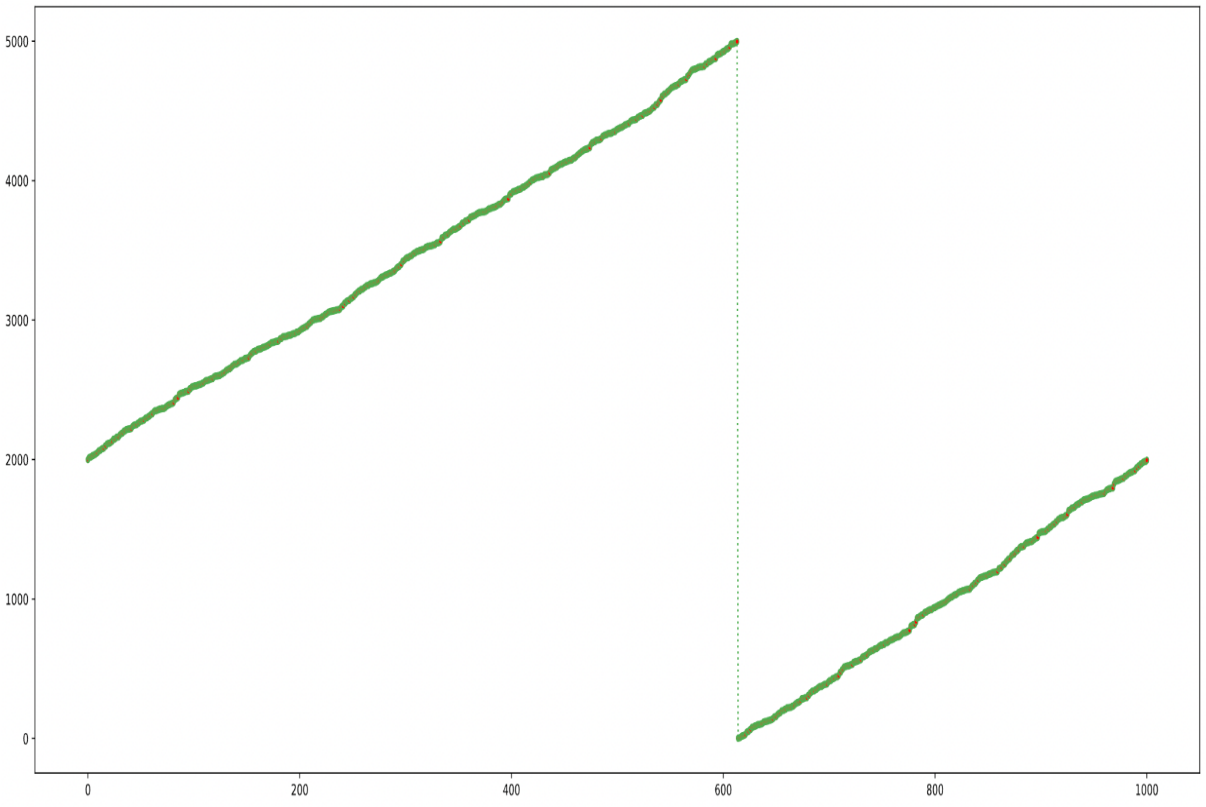
4. C-SCAN



5. LOOK



6. C-LOOK



- Advantages and dis-advantages of each algorithm
  1. FCFS
     - Advantages
       - Every request gets a fair chance.
       - No starvation of any request.
     - Dis-advantages
       - No optimisation for seek time.
  2. SSTF
     - Advantages
       - High throughput.
       - Av. seek time is very less.
     - Dis-advantages
       - There's an overhead for calculating seek time in each direction in advance.
       - Can cause starvation.
  3. SCAN
     - Advantages
       - Low variance of response time.
       - Av. seek time is very less.
     - Dis-advantages
       - Long waiting time for some requests.
  4. C-SCAN
     - Advantages
       - Uniform waiting time.
       - Av. seek time is less.
     - Dis-advantages
       - More seek movements than SCAN.
  5. LOOK
     - Advantages
       - Better response than SCAN.
       - No starvation of any request.
     - Dis-advantages
       - Overhead of finding start and end of the requests.
  6. C-LOOK
     - Advantages
       - Low variance in the response and waiting time.
       - No starvation of any request.
     - Dis-advantages
       - Overhead of finding start and end of the requests.

- Inference: Best algorithm: **SCAN** or **SSTF**.