CS342 –Project 4

Report

Kübra Nur Güzel- 21400946

Arif Can Terzioğlu- 21302061

Environment

Our main system both work on Windows 10 OS. On the hardware side, our computer processers are 4 cores Intel i7 4700HQ 2.Ghz series. Ram amounts are both 16GB. We have same generation of notebook of ASUS. We installed Ubuntu 16.04 64bit via virtual machine. Arif allocated two core of his computer processor. Kübra allocated one core of her computer processor. 5000 megabyte is allocated for main memory.

Design Stage

We started to do project a little bit late but it gave us opportunity to get benefit from piazza post. As it can be seen from piazza biggest ambiguity was request arrival time and when we will the serve the new coming request while head is moving and etc. Also, when we looked at the clarification, we saw that we needed to pay attention to timing in other word simulate the time.

To achieve this we planned to use two different data structure that one holds the whole input file and one holds the requests until current time.

Basically what we did this on all algorithms as time moves on, we added the coming request to queue. However, we did not serve the request coming directly while another request is begin served.

In a loop:

> Add newly coming requests to request queue.
>
> If there are no request in the queue and if we know there is a remaining request in the file we basically increase the time.
>
> If both there are no request in request queue and file, we terminate.
>
> Do the necessary calculations based on algorithms.

FCFS: The straight forward one among the algorithms. Directly serve the request coming first.

SSTF: It was like finding the minimum in array problem with respect to cylinder number. New added request situation was an important factor here.

LOOK: Different than above two algorithms, we have a direction here. For direction right we need to find the closest request to head in other words closest greater cylinder number to head and while going left find the closest request to head in other words closest smaller cylinder number to head. Newly added request are handled after current request being served is finished.

CLOOK: Different than Look, we have only one direction. When we come to the end of the right direction, we need to find furthest point to the head on the left in other words smallest number to head. Important point is here that reversing the direction and goes the beginning (to the left-most request) without serving (but still moving the disk and spending time).

We started disk head at cylinder number 1.

Implementation Stage

Choosing the proper data structure for each algorithm was the first thing for each algorithm. For FCFS, queue structure was very straightforward. For SSTF, LOOK and CLOOK, we needed to do a lot retrieval we can achieve this via array in O (1) time. If we used linked list structure, since complexity may up to O (n), it may decrease the efficiency. In addition, we keep a data array for mean and standard deviation. Also, since first request of each algorithm will have a waiting time of 0, we divided n-1 for standard deviation.

We start implementation by reading the txt file to our queue. For FCFS, the order of input file is applied since it comes to us sorted w.r.t to arrival time. Wait time for each request is calculated, and then head and current time is recalculated. In SSTF, we write a "findMin" method that finds the closest request to head. For look, we added a direction variable that if there is no remaining request in current direction, it swaps the directions via if-else structure. In LOOK algorithm, we wrote to methods that are "findNextHigher" and "findNextLower" for finding the closest request in respective direction. In CLOOK algorithm, we used modification of findNexHigher method since we have only one direction. It just changes the current head value to smallest request number on the left side. Rest is kind of same with LOOK algorithm.

Challenges during Implementation

One of the challenges we encountered was the synchronization of request being served and newly coming request during that time. Since request should not be added to the queue earlier than its arrival time, we put a lot effort on make request-serving mechanism with current time.

Each algorithm requires different search method for request will be served. Designing search mechanism for LOOK and CLOOK required overthinking a bit. We did a lot debugging by printing the data values for mean and order of cylinders served. Some examples from the piazza were used to validate our results. They helped a lot.

While dealing with standard deviation, even though we were finding the total head movement, order and mean correct, we could not find the standard deviation properly. We solved this problem by change divider n to n-1. We predicted that it occurs because first request has waiting time 0 and has no effect. Also, making precision double provided exact match with given example output.

Another interesting point was that we never used N parameter. It could be used but we did not prefer to implement with N.

Testing

During the testing stage, we used sample test case provided on the piazza. Also we used sample input file on the project 4 assignment paper. For that result, we calculated its values by hand and compare the result with them. Sometimes on piazza, some samples and cases are stated on some topic. We always compare these cases with our results for consistency.

Experiment

We conducted experiment in two ways that are one for increasing number of request and one for a lot of different sequence of requests for a given number of requests.

Methodology

On the sequence change experiment, each value is measured three times and their average values are taken. Number of request is fixed here.

In increasing number of request experiment, for each number, experiment conducted three times by changing the order a little bit and gets their average value.

After the graphics, results and conclusion are explained.

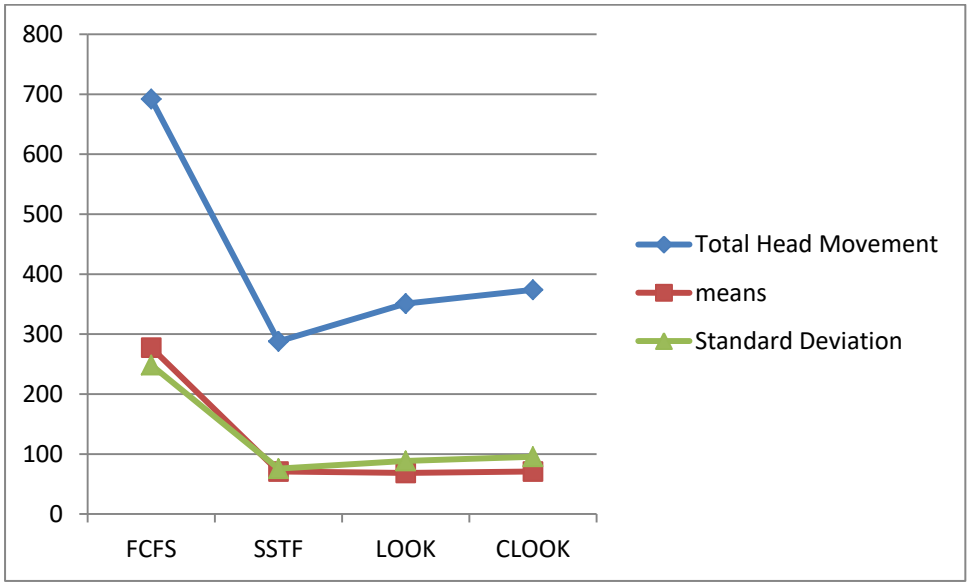
Inputs

Increasing number of request: 9, 18, 27

Different sequence of requests for a given number of requests.

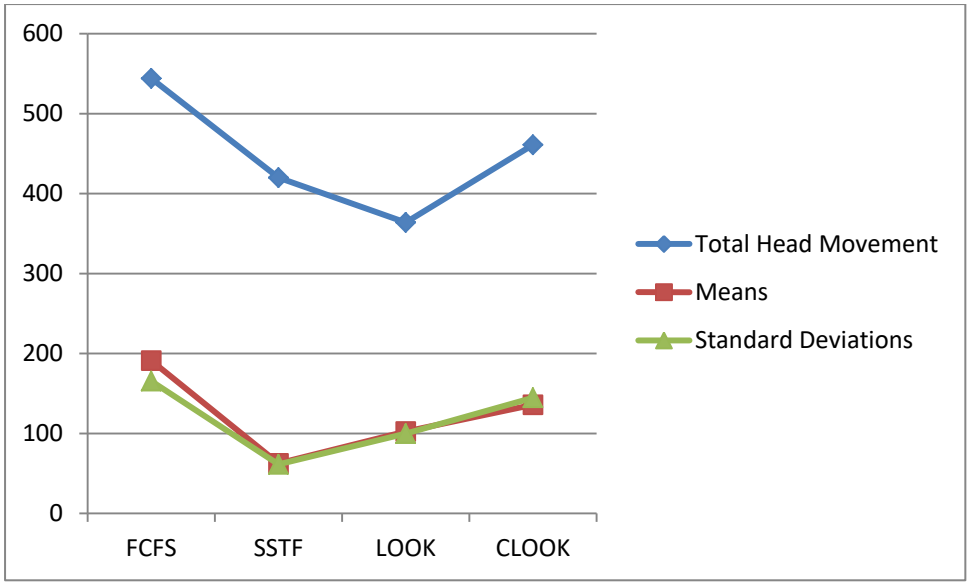
Results

Changing Sequence Experiment



Axis X: Algorithms Axis Y: Time

	total	mean	SD
FCFS	692	277.555	248.579
SSTF	288	71.11	75.836
LOOK	351	68.555	88.701
CLOOK	374	71.11	95.49



Axis X: Algorithms Axis Y: Amount

	Total	Mean	SD
FCFS	544	191.111	165.617
SSTF	420	62.66	61.49
LOOK	364	102.44	99.8
CLOOK	461	135.88	144.67

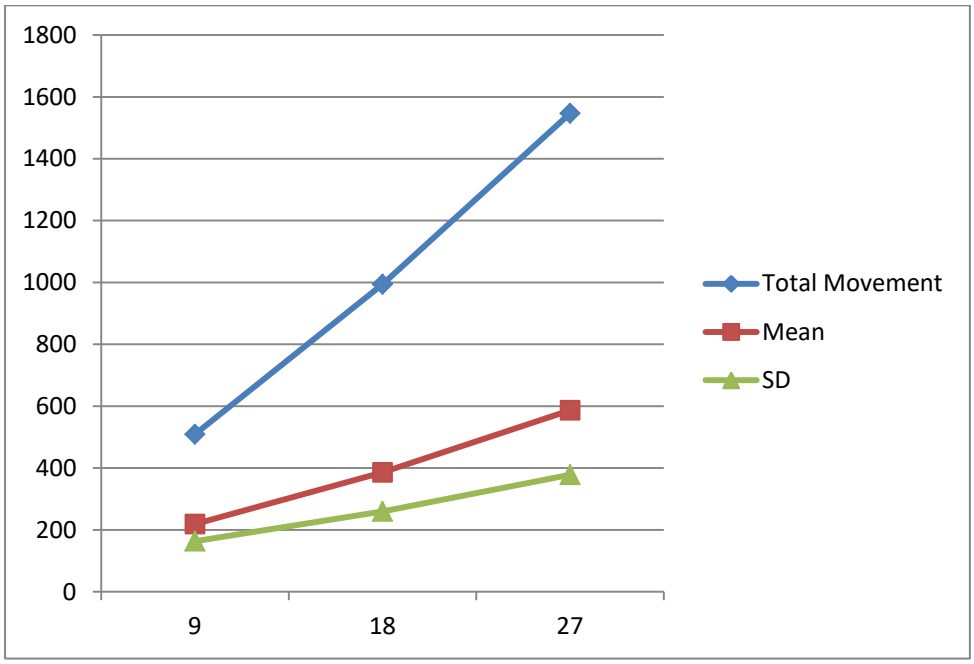
Axis X: Algorithms Axis Y: Amount

	Total	Mean	SD
FCFS	575	230.666	186.29
SSTF	310	83.66	79.95
LOOK	364	130.333	98.658
CLOOK	487	215.111	171.259

When we change the sequence of the cylinder in a fixed numbered input file, we can see that in each graphics, orientation of lines is similar. FCFS remains highest. SSTF and LOOK seems more prone changes on sequence. Also, CLOOK seems consistent also. As we change the sequence, standard deviation of SSTF, LOOK and CLOOK are increased. FCFS standard total movement is more likely change since it is heavily based on the order coming request.

Increasing Number of Requests

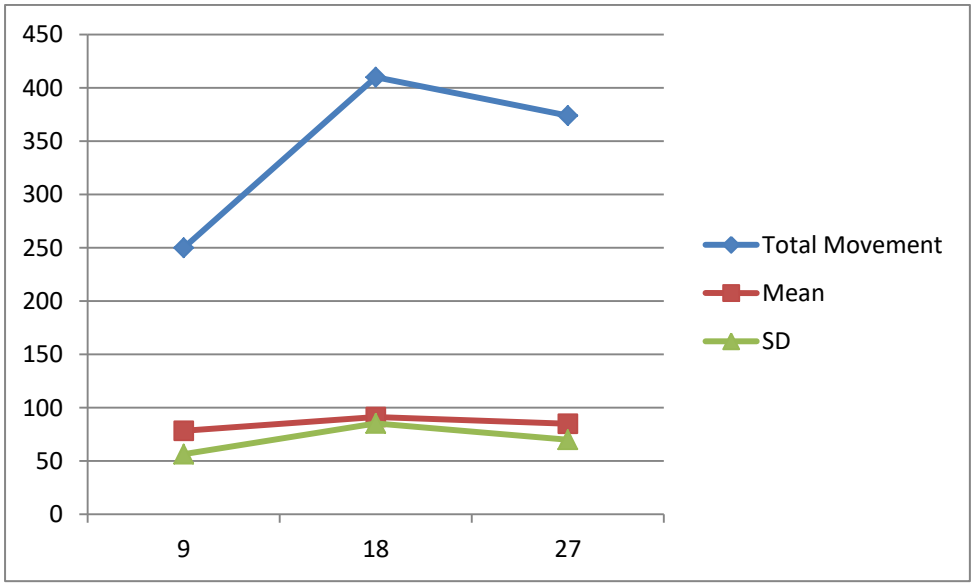
FCFS



Axis X = Number of Request Axis Y: Time

In FCFS algorithm, it is expected that as the number of request increases, all of them also increases such as total movement, mean and SD. Increase is higher than other algorithms

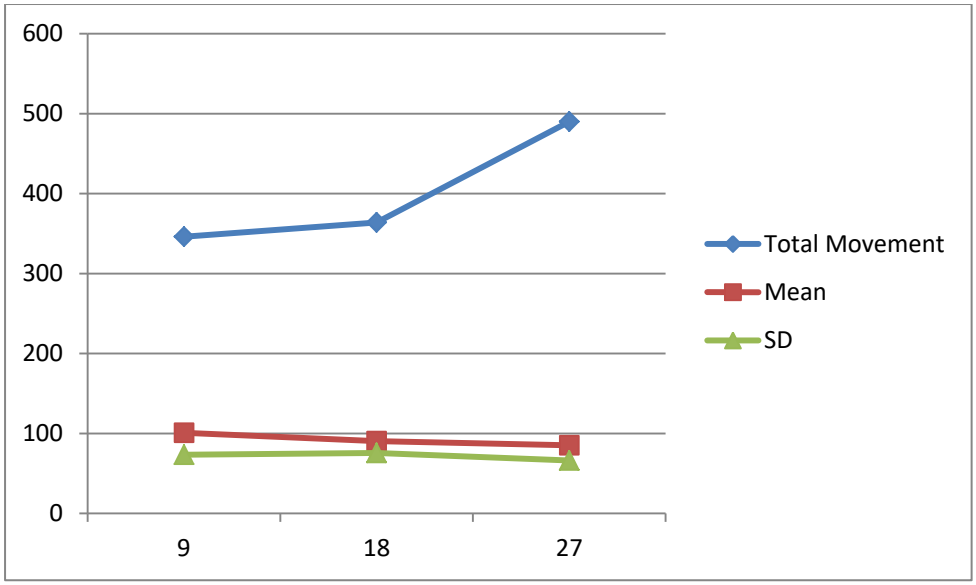
SSTF



Axis X = Number of Request Axis Y: Time

We can see that in SSTF if random sequence comes in a well-positioned manner, we may get better results even though number of request increases.

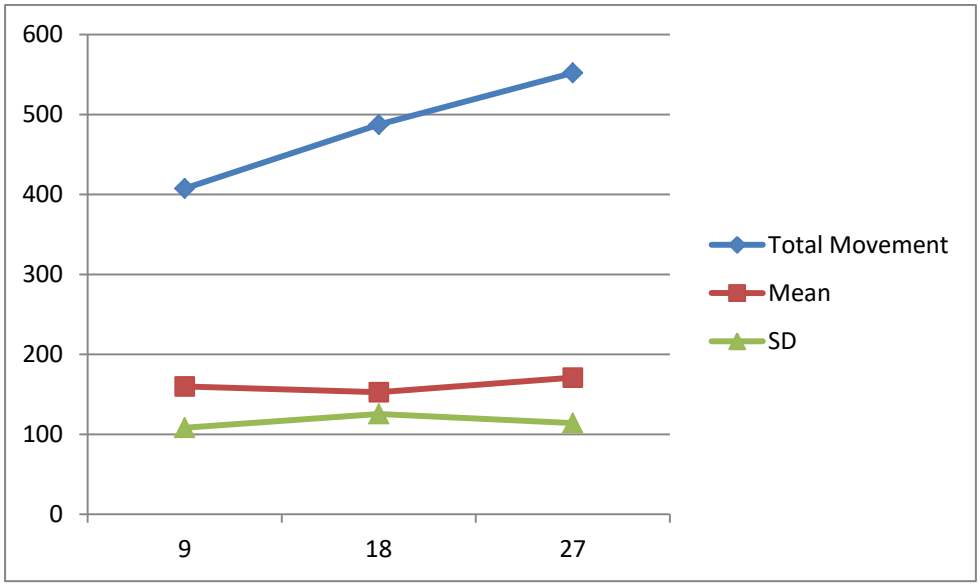
LOOK



Axis X = Number of Request Axis Y: Time

Compared to FCFS, Look algorithm seemed to me more efficient and increase is not that sharp in total head movement.

CLOOK



Axis X = Number of Request Axis Y: Time

Clook algorithm also algorithm seemed to me more efficient and increase is not that sharp in total head movement. Normally, Clook is expected to perform better than look but in our experiment we observed the otherwise.

Table Values

9	Total	Mean	SD	Total	Mean	SD	Avg Total	Avg Mean	Avg SD
FCFS	545	255.66	183.63	473	182	142.75	509	218.83	163.19
SSTF	286	96.88	61.5313	214	59.888	51.3649	250	78.384	56.4481
LOOK	364	146.88	97.745	328	54.77	49.07	346	100.825	73.4075
CLOOK	487	265.222	167.569	328	54.77	49.07	407.5	159.996	108.3195

18	Total	Mean	SD	Total	Mean	SD	Avg Total	Avg Mean	Avg SD
FCFS	1067	423.83	286.56	921	347.66	232.433	994	385.745	259.4965
SSTF	456	97.66	100.005	364	84.94	70.55	410	91.3	85.2775
LOOK	364	96	80.8637	364	84.944	70.5586	364	90.472	75.71115
CLOOK	490	165.11	130.085	485	140.44	120.87	487.5	152.775	125.4775

27	Total	Mean	SD	Total	Mean	SD	Avg Total	Avg Mean	Avg SD
FCFS	1465	562.8518	373.65	1628	610.85	383.126	1546.5	586.8509	378.388
SSTF	374	84.5925	71.2877	374	85.44	68.5022	374	85.01625	69.89495
LOOK	490	84.814	67.8902	490	85.66	64.7426	490	85.237	66.3164
CLOOK	490	167.777	107.3639	614	174.259	120.8799	552	171.018	114.1219