

**CS 4310 Operating Systems**  
**Project #1 Simulating Job Scheduler and Performance Analysis**

**Due: 6/23**  
(Total: 100 points)

**Student Name:** Alan Huang

**Date:** 06/12/2020

***Important:***

- Please read this document completely before you start coding.
- Also, please read the submission instructions (provided at the end of this document) carefully before submitting the project.

***Project #1 Description:***

Simulating Job Scheduler of the Operating Systems by programming the following four scheduling algorithms that we covered in the class:

- a. First-Come-First-Serve (FCFS)
- b. Shortest-Job-First (SJF)
- c. Round-Robin with Time Slice = 2 (RR-2)
- d. Round-Robin with Time Slice = 5 (RR-5)

You can use either Java or your choice of programming language for the implementation. The objective of this project is to help student understand how above four job scheduling algorithms operates by implementing the algorithms, and conducting a performance analysis of them based on the performance measure of their average turnaround times (of all jobs) for each scheduling algorithm using multiple inputs. Output the details of each algorithm's execution. You need to show which jobs are selected at what times as well as their starting and stopping burst values. You can choose your display format, for examples, you can display the results of each in *Schedule Table* or *Gantt Chart* format (as shown in the class notes). The project will be divided into three parts (phases) to help you to accomplish above tasks in a systematic and scientific fashion: Design and Testing, Implementation, and Performance Analysis.

The program will read process burst times from a file (job.txt) – this file will be generated by you. Note that you need to generate multiple testing cases (with inputs of 5 jobs, 10 jobs and 15 jobs). A sample input file of five jobs is given as follows (burst time in ms):

```
[Begin of job.txt]
Job1
7
Job2
18
Job3
10
Job4
4
Job5
12
[End of job.txt]
```

Note: you can assume that

- (1) There are no more than 30 jobs in the input file (job.txt).
- (2) Processes arrive in the order they are read from the file for FCFS, RR-2 and RR-5.
- (3) All jobs arrive at time 0.
- (4) FCFS use the order of the jobs, Job1, Job2, Job3, ...

You can implement the algorithms in your choice of data structures based on the program language of your choice. Note that you always try your best to give the most efficient program for each problem. The size of the input will be limited to be within 30 jobs.

***Submission Instructions:***

- *turn in the following @[blackboard.cpp.edu](https://blackboard.cpp.edu) after the completion of all three parts, part 1, part 2 and part 3*
  - (1) four program files (your choice of programming language with proper documentation)*
  - (2) this document (complete all the answers)*

## Part1

### Design & Testing (30 points)

- a. Design the program by providing pseudocode or flowchart for each CPU scheduling algorithm.

```
<
FCFS(LinkedList JobList)
{
    Int Total = JobList size
    Int StartTime = 0
    Int EndTime = 0
    Float TurnaroundTime = 0

    Print schedule table header first

    While(JobList is not empty)
    {
        EndTime = First node's time;
        Print schedule table rows result with StartTime, EndTime, and completed time
          which is same as EndTime

        Turnaround Time = TurnaroundTime + EndTime
        StartTime = Endtime

        Remove head of list
    }
    Print average turnaround time which is TurnaroundTime / Total
}
```

```
SJF(LinkedList JobList)
{
    First sort the list with ascending order

    Int Total = JobList size
    Int StartTime = 0
    Int EndTime = 0
    Float TurnaroundTime = 0

    Print schedule table header first

    While(JobList is not empty)
    {
        EndTime = First node's time;
        Print schedule table rows result with StartTime, EndTime, and completed time
          which is same as EndTime

        Turnaround Time = TurnaroundTime + EndTime
        StartTime = Endtime

        Remove head of list
    }
}
```

```

    }
    Print average turnaround time which is TurnaroundTime / Total
}

RoundRobin(LinkedList JobList, int slices)
{
    Int Total = JobList size
    Int StartTime = 0
    Int EndTime = 0
    Float TurnaroundTime = 0

    If slice = 2
        Print schedule table header for Round Robin 2
    If slice = 5
        Print schedule table header for Round Robin 5

    While(JobList is not empty)
    {
        If first node time – slices <= 0
            EndTime = EndTime + first node's time

            Print schedule table rows result with StartTime, EndTime, and complete
            time which is same as EndTime

            TurnaroundTime = TurnaroundTime + end Time

            StartTime = EndTime

        Else
            EndTime = endTime + slices

            Print schedule table rows result with StartTime, EndTime, but no
            completed time

            StartTime = EndTime

            Since the head is not finished so add it to the last node, and remove the
            first node

    }

    Print average turnaround time which is TurnaroundTime / Total
}

>

```

- b. Design the program correctness testing cases. Give at least 3 testing cases to test your program, and give the expected correct average turnaround time (for each testing case) in order to test the correctness of each algorithm.

<Complete the following table>

Testing case #	Input (table of jobs with its job# and length	Expected output for FCFS (√ if Correct after testing in Part 3)	Expected output for SJF (√ if Correct after testing in Part 3)	Expected output for RR-2 (√ if Correct after testing in Part 3)	Expected output for RR-5 (√ if Correct after testing in Part 3)
1 (5 jobs)	< Job1 9 Job2 6 Job3 3 Job4 12 Job5 8 >	< Expected = 22 >  √	< Expected = 18.6 >  √	< Expected = 28.6 >  √	< Expected = 28.4 >  √
2 (10 jobs)	< Job1 9 Job2 6 Job3 3 Job4 12 Job5 8 Job6 6 Job7 16 Job8 18 Job9 4 Job10 10 >	< Expected = 46.6 >  √	< Expected = 37.5 >  √	< Expected = 61.7 >  √	< Expected = 61.4 >  √

3 (15 jobs)	<	<	<	<	<
	Job1	Expected	Expected	Expected	Expected
	9	= 67.667	= 47.6	= 81.8	= 83.4
	Job2	>	>	>	>
	6				
	Job3	√	√	√	√
	3				
	Job4				
	12				
	Job5				
	8				
	Job6				
	6				
	Job7				
	16				
	Job8				
	18				
	Job9				
	4				
	Job10				
	10				
	Job11				
	4				
	Job12				
	1				
	Job13				
	9				
	Job14				
	16				
	Job15				
	6				
	>				

- c. Design testing strategy for the programs. Discuss about how to generate and structure the randomly generated inputs for experimental study later in Part 3.

*Hint 1: To study the performance evaluation of the four job scheduling algorithms, this project will use three different input sizes, 5 jobs, 10 jobs and 15 jobs. It is the easiest to use a random number generator for generating the inputs. Note that you need to decide the maximum value of job length (use at least 20). However, student should store each data set in various sizes and use the same data set for each job scheduling algorithm.*

*The performance of average Turnaround Time of each input data size (5 jobs, 10 jobs and 15 jobs) can be calculated after an experiment is conducted in 20 trail (with 20 input sets of jobs). We can denote the results as the set  $X$  which contains the 20 computed Turnaround Times of 20 trails, where  $X = \{x_1, x_2, x_3 \dots x_{20}\}$ , from the simulator.*

*For each data size (5 jobs, 10 jobs and 15 jobs):*

$$\text{Average Turnaround Time} = \frac{\sum_{i=1}^{20} x_i}{20}$$

*The student should decide the maximum value of the job length (at least 20).*

My testing strategy for the programs, I first create a method called randomJob and pass a Linked list and desire size to it. The randomJob method will use a for loop with Math.random to create random inputs. Second, I let the program ask the user how many random jobs they want from 5, 10, and 15, then the program will enter a for loop which will run 20 times. For each time the for loop will create a random job list and passed it to all three algorithms (I only have 1 Round Robin) to test the time it needs. Each algorithm's method will help record each trial's time and the sum of 20 trial's time. I also use a 2d array to store each algorithm's time, so I can call it later for display. After each run time the program will reset each algorithm's trial. After that, the program will display each trial's average, and the average turnaround time, then it will reset the average turnaround time and ready for the next testing. >

**Part 2**  
**Implementation (30 points)**

- a. Code each program based on the design (pseudocode or flow chart) in Part 1(a).

<Generate four programs and stored them in four files, needed to be submitted>

- b. Document the program appropriately.

<Generate documentation inside the four program files>

- c. Test you program using the designed testing input data given in the table in Part 1(b), Make sure each program generates the correct answer by marking a “√” if it is correct for each testing case for each program column in the table. Repeat the process of debugging if necessary.

<Complete the four columns of the four algorithms in the table @Part 1(b)>

- d. For each program, capture a screen shot of the execution (Compile&Run) using the testing case in Part 1(b) to show how this program works properly

**5 Job**

**(a) First Come First Service**

Job#	Start time	End Time	Job Completion
Job1	0	9	Job1 completed @ 9
Job2	9	15	Job2 completed @ 15
Job3	15	18	Job3 completed @ 18
Job4	18	30	Job4 completed @ 30
Job5	30	38	Job5 completed @ 38

FCFS Average Turnaround Time is = 22.000

**(b) Shortest Job First**

Job#	Start time	End Time	Job Completion
Job3	0	3	Job3 completed @ 3
Job2	3	9	Job2 completed @ 9
Job5	9	17	Job5 completed @ 17
Job1	17	26	Job1 completed @ 26
Job4	26	38	Job4 completed @ 38

SJF Average Turnaround Time is = 18.600



(c) Round Robin with Time Slice = 2

Job#	Start time	End Time	Job Completion
Job1	0	2	
Job2	2	4	
Job3	4	6	
Job4	6	8	
Job5	8	10	
Job1	10	12	
Job2	12	14	
Job3	14	15	Job3 completed @ 15
Job4	15	17	
Job5	17	19	
Job1	19	21	
Job2	21	23	Job2 completed @ 23
Job4	23	25	
Job5	25	27	
Job1	27	29	
Job4	29	31	
Job5	31	33	Job5 completed @ 33
Job1	33	34	Job1 completed @ 34
Job4	34	36	
Job4	36	38	Job4 completed @ 38

Average Turnaround Time is = 28.600

(d) Round Robin with Time Slice = 5

Job#	Start time	End Time	Job Completion
Job1	0	5	
Job2	5	10	
Job3	10	13	Job3 completed @ 13
Job4	13	18	
Job5	18	23	
Job1	23	27	Job1 completed @ 27
Job2	27	28	Job2 completed @ 28
Job4	28	33	
Job5	33	36	Job5 completed @ 36
Job4	36	38	Job4 completed @ 38

Average Turnaround Time is = 28.400

10 Job

(a) First Come First Service

Job#	Start time	End Time	Job Completion
Job1	0	9	Job1 completed @ 9
Job2	9	15	Job2 completed @ 15
Job3	15	18	Job3 completed @ 18
Job4	18	30	Job4 completed @ 30
Job5	30	38	Job5 completed @ 38
Job6	38	44	Job6 completed @ 44
Job7	44	60	Job7 completed @ 60
Job8	60	78	Job8 completed @ 78
Job9	78	82	Job9 completed @ 82
Job10	82	92	Job10 completed @ 92

FCFS Average Turnaround Time is = 46.600

(b) Shortest Job First

Job#	Start time	End Time	Job Completion
Job3	0	3	Job3 completed @ 3
Job9	3	7	Job9 completed @ 7
Job2	7	13	Job2 completed @ 13
Job6	13	19	Job6 completed @ 19
Job5	19	27	Job5 completed @ 27
Job1	27	36	Job1 completed @ 36
Job10	36	46	Job10 completed @ 46
Job4	46	58	Job4 completed @ 58
Job7	58	74	Job7 completed @ 74
Job8	74	92	Job8 completed @ 92

SJF Average Turnaround Time is = 37.500

(c) Round Robin with Time Slice = 2			
Job#	Start time	End Time	Job Completion
Job1	0	2	
Job2	2	4	
Job3	4	6	
Job4	6	8	
Job5	8	10	
Job6	10	12	
Job7	12	14	
Job8	14	16	
Job9	16	18	
Job10	18	20	
Job1	20	22	
Job2	22	24	
Job3	24	25	Job3 completed @ 25
Job4	25	27	
Job5	27	29	
Job6	29	31	
Job7	31	33	
Job8	33	35	
Job9	35	37	Job9 completed @ 37
Job10	37	39	
Job1	39	41	
Job2	41	43	Job2 completed @ 43
Job4	43	45	
Job5	45	47	
Job6	47	49	Job6 completed @ 49
Job7	49	51	
Job8	51	53	
Job10	53	55	
Job1	55	57	
Job4	57	59	
Job5	59	61	Job5 completed @ 61
Job7	61	63	
Job8	63	65	
Job10	65	67	
Job1	67	68	Job1 completed @ 68
Job4	68	70	

Job7	70	72	
Job8	72	74	
Job10	74	76	Job10 completed @ 76
Job4	76	78	Job4 completed @ 78
Job7	78	80	
Job8	80	82	
Job7	82	84	
Job8	84	86	
Job7	86	88	Job7 completed @ 88
Job8	88	90	
Job8	90	92	Job8 completed @ 92

Average Turnaround Time is = 61.700

(d) Round Robin with Time Slice = 5

Job#	Start time	End Time	Job Completion
Job1	0	5	
Job2	5	10	
Job3	10	13	Job3 completed @ 13
Job4	13	18	
Job5	18	23	
Job6	23	28	
Job7	28	33	
Job8	33	38	
Job9	38	42	Job9 completed @ 42
Job10	42	47	
Job1	47	51	Job1 completed @ 51
Job2	51	52	Job2 completed @ 52
Job4	52	57	
Job5	57	60	Job5 completed @ 60
Job6	60	61	Job6 completed @ 61
Job7	61	66	
Job8	66	71	
Job10	71	76	Job10 completed @ 76
Job4	76	78	Job4 completed @ 78
Job7	78	83	
Job8	83	88	
Job7	88	89	Job7 completed @ 89
Job8	89	92	Job8 completed @ 92

Average Turnaround Time is = 61.400

15 Job

(a) First Come First Service

Job#	Start time	End Time	Job Completion
Job1	0	9	Job1 completed @ 9
Job2	9	15	Job2 completed @ 15
Job3	15	18	Job3 completed @ 18
Job4	18	30	Job4 completed @ 30
Job5	30	38	Job5 completed @ 38
Job6	38	44	Job6 completed @ 44
Job7	44	60	Job7 completed @ 60
Job8	60	78	Job8 completed @ 78
Job9	78	82	Job9 completed @ 82
Job10	82	92	Job10 completed @ 92
Job11	92	96	Job11 completed @ 96
Job12	96	97	Job12 completed @ 97
Job13	97	106	Job13 completed @ 106
Job14	106	122	Job14 completed @ 122
Job15	122	128	Job15 completed @ 128

FCFS Average Turnaround Time is = 67.667

Job#	Start time	End Time	Job Completion
Job12	0	1	Job12 completed @ 1
Job3	1	4	Job3 completed @ 4
Job9	4	8	Job9 completed @ 8
Job11	8	12	Job11 completed @ 12
Job2	12	18	Job2 completed @ 18
Job6	18	24	Job6 completed @ 24
Job15	24	30	Job15 completed @ 30
Job5	30	38	Job5 completed @ 38
Job1	38	47	Job1 completed @ 47
Job13	47	56	Job13 completed @ 56
Job10	56	66	Job10 completed @ 66
Job4	66	78	Job4 completed @ 78
Job7	78	94	Job7 completed @ 94
Job14	94	110	Job14 completed @ 110
Job8	110	128	Job8 completed @ 128

SJF Average Turnaround Time is = 47.600

(c) Round Robin with Time Slice = 2

Job#	Start time	End Time	Job Completion
Job1	0	2	
Job2	2	4	
Job3	4	6	
Job4	6	8	
Job5	8	10	
Job6	10	12	
Job7	12	14	
Job8	14	16	
Job9	16	18	
Job10	18	20	
Job11	20	22	
Job12	22	23	Job12 completed @ 23
Job13	23	25	
Job14	25	27	
Job15	27	29	
Job1	29	31	
Job2	31	33	
Job3	33	34	Job3 completed @ 34
Job4	34	36	
Job5	36	38	
Job6	38	40	
Job7	40	42	
Job8	42	44	
Job9	44	46	Job9 completed @ 46
Job10	46	48	
Job11	48	50	Job11 completed @ 50
Job13	50	52	
Job14	52	54	
Job15	54	56	
Job1	56	58	
Job2	58	60	Job2 completed @ 60
Job4	60	62	
Job5	62	64	
Job6	64	66	Job6 completed @ 66
Job7	66	68	
Job8	68	70	
Job10	70	72	
Job13	72	74	
Job14	74	76	

<

Job1	78	80	
Job4	80	82	
Job5	82	84	Job5 completed @ 84
Job7	84	86	
Job8	86	88	
Job10	88	90	
Job13	90	92	
Job14	92	94	
Job1	94	95	Job1 completed @ 95
Job4	95	97	
Job7	97	99	
Job8	99	101	
Job10	101	103	Job10 completed @ 103
Job13	103	104	Job13 completed @ 104
Job14	104	106	
Job4	106	108	Job4 completed @ 108
Job7	108	110	
Job8	110	112	
Job14	112	114	
Job7	114	116	
Job8	116	118	
Job14	118	120	
Job7	120	122	Job7 completed @ 122
Job8	122	124	
Job14	124	126	Job14 completed @ 126
Job8	126	128	Job8 completed @ 128

Average Turnaround Time is = 81.800

(d) Round Robin with Time Slice = 5

Job#	Start time	End Time	Job Completion
Job1	0	5	
Job2	5	10	
Job3	10	13	Job3 completed @ 13
Job4	13	18	
Job5	18	23	
Job6	23	28	
Job7	28	33	
Job8	33	38	
Job9	38	42	Job9 completed @ 42
Job10	42	47	
Job11	47	51	Job11 completed @ 51
Job12	51	52	Job12 completed @ 52
Job13	52	57	
Job14	57	62	
Job15	62	67	
Job1	67	71	Job1 completed @ 71
Job2	71	72	Job2 completed @ 72
Job4	72	77	
Job5	77	80	Job5 completed @ 80
Job6	80	81	Job6 completed @ 81
Job7	81	86	
Job8	86	91	
Job10	91	96	Job10 completed @ 96
Job13	96	100	Job13 completed @ 100
Job14	100	105	
Job15	105	106	Job15 completed @ 106
Job4	106	108	Job4 completed @ 108
Job7	108	113	
Job8	113	118	
Job14	118	123	
Job7	123	124	Job7 completed @ 124
Job8	124	127	Job8 completed @ 127
Job14	127	128	Job14 completed @ 128

Average Turnaround Time is = 83.400



By now, four working programs are created and ready for experimental study in the next part, Part 3.

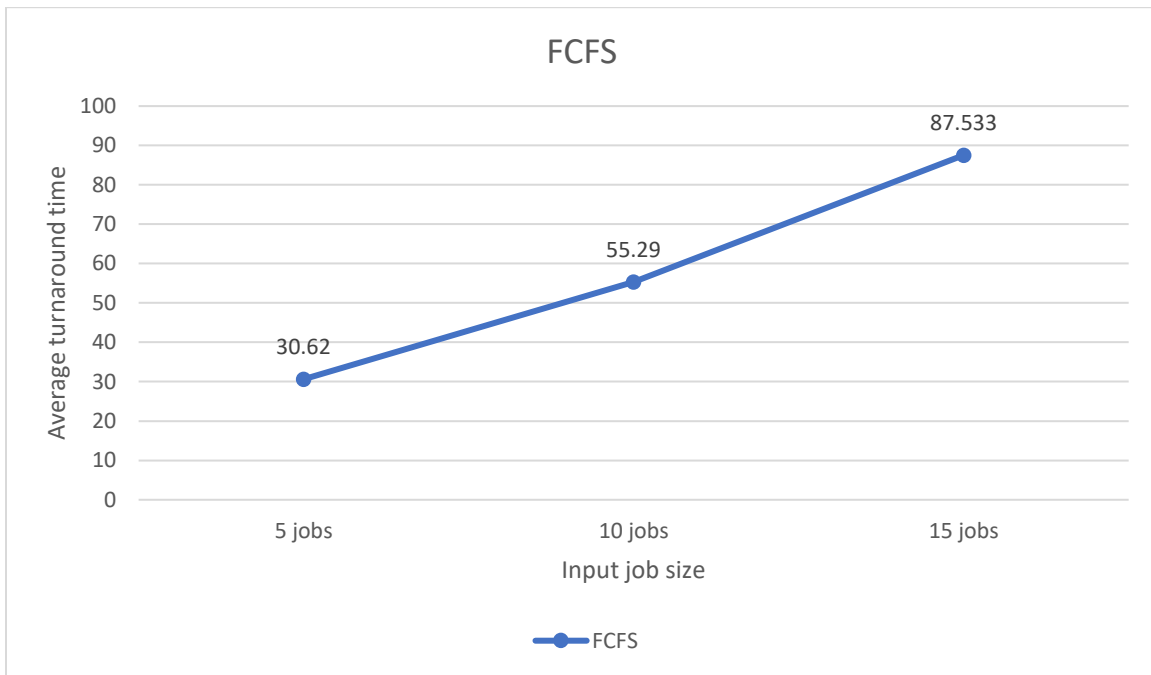
**Part 3**  
**Performance Analysis (40 points)**

- a. Run each program with the designed randomly generated input data given in Part 1(c).  
Generate a table for all the experimental results for performance analysis as follows.

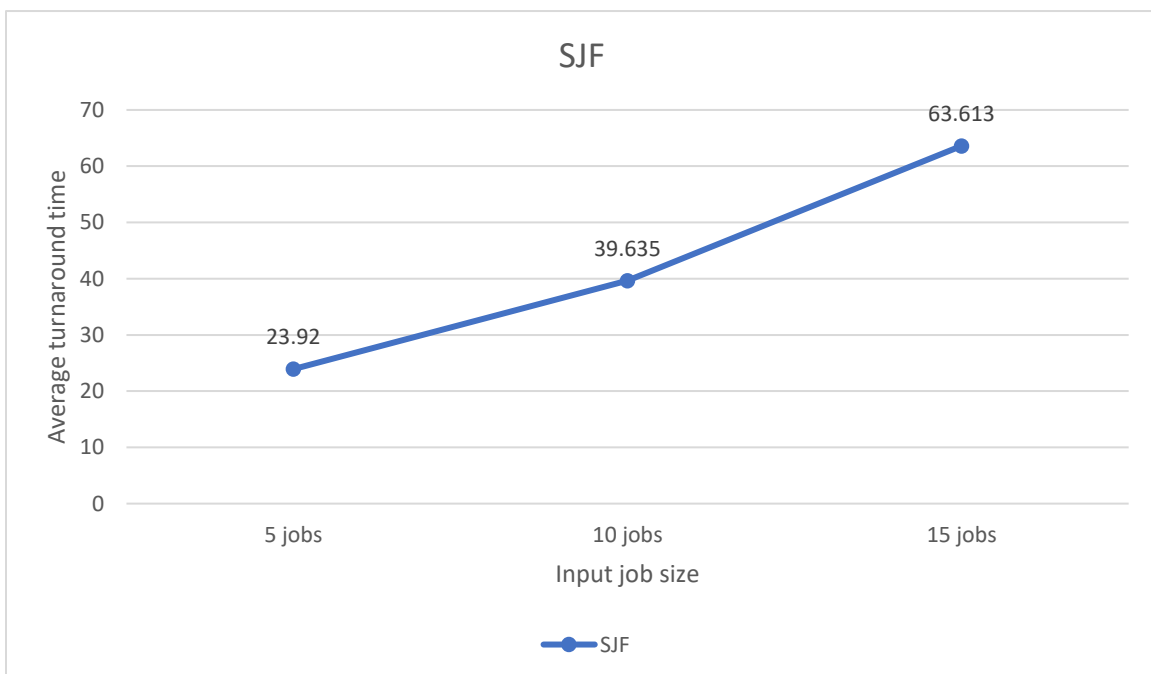
Input Size n jobs	Average of average turnaround times (FCFS Program)	Average of average turnaround times (SJF Program)	Average of average turnaround times (RR-2)	Average of average turnaround times (RR-5)
5 jobs	FCFS trial 1 is 32.200 FCFS trial 2 is 31.800 FCFS trial 3 is 28.800 FCFS trial 4 is 34.800 FCFS trial 5 is 30.400 FCFS trial 6 is 27.200 FCFS trial 7 is 40.600 FCFS trial 8 is 35.800 FCFS trial 9 is 30.800 FCFS trial 10 is 25.800 FCFS trial 11 is 27.400 FCFS trial 12 is 44.200 FCFS trial 13 is 19.600 FCFS trial 14 is 17.200 FCFS trial 15 is 23.000 FCFS trial 16 is 20.400 FCFS trial 17 is 33.000 FCFS trial 18 is 45.600 FCFS trial 19 is 31.600 FCFS trial 20 is 32.200  FCFS average over 20 trials is 30.620	SJF trial 1 is 23.600 SJF trial 2 is 25.200 SJF trial 3 is 22.800 SJF trial 4 is 24.400 SJF trial 5 is 25.600 SJF trial 6 is 19.400 SJF trial 7 is 29.600 SJF trial 8 is 26.800 SJF trial 9 is 20.600 SJF trial 10 is 22.800 SJF trial 11 is 24.800 SJF trial 12 is 44.200 SJF trial 13 is 9.200 SJF trial 14 is 12.800 SJF trial 15 is 17.200 SJF trial 16 is 11.000 SJF trial 17 is 30.600 SJF trial 18 is 37.800 SJF trial 19 is 25.200 SJF trial 20 is 24.800  SJF average over 20 trials is 23.920	RR2 trial 1 is 34.800 RR2 trial 2 is 37.200 RR2 trial 3 is 32.800 RR2 trial 4 is 36.200 RR2 trial 5 is 38.400 RR2 trial 6 is 30.400 RR2 trial 7 is 47.200 RR2 trial 8 is 41.200 RR2 trial 9 is 30.600 RR2 trial 10 is 36.400 RR2 trial 11 is 40.200 RR2 trial 12 is 69.400 RR2 trial 13 is 12.400 RR2 trial 14 is 19.400 RR2 trial 15 is 25.200 RR2 trial 16 is 17.600 RR2 trial 17 is 48.600 RR2 trial 18 is 58.200 RR2 trial 19 is 38.400 RR2 trial 20 is 37.600  RR2 average over 20 trials is 36.610	RR5 trial 1 is 38.600 RR5 trial 2 is 37.200 RR5 trial 3 is 33.800 RR5 trial 4 is 39.600 RR5 trial 5 is 40.400 RR5 trial 6 is 30.600 RR5 trial 7 is 49.000 RR5 trial 8 is 41.200 RR5 trial 9 is 30.600 RR5 trial 10 is 32.400 RR5 trial 11 is 37.400 RR5 trial 12 is 65.200 RR5 trial 13 is 15.200 RR5 trial 14 is 17.000 RR5 trial 15 is 24.200 RR5 trial 16 is 19.200 RR5 trial 17 is 46.000 RR5 trial 18 is 58.800 RR5 trial 19 is 35.800 RR5 trial 20 is 36.800  RR5 average over 20 trials is 36.450
10 jobs	FCFS trial 1 is 57.300 FCFS trial 2 is 56.800 FCFS trial 3 is 59.700 FCFS trial 4 is 51.700 FCFS trial 5 is 57.700 FCFS trial 6 is 58.200 FCFS trial 7 is 58.300 FCFS trial 8 is 55.200 FCFS trial 9 is 64.900 FCFS trial 10 is 50.100 FCFS trial 11 is 46.000 FCFS trial 12 is 74.200 FCFS trial 13 is 75.400 FCFS trial 14 is 51.800 FCFS trial 15 is 36.400 FCFS trial 16 is 54.200 FCFS trial 17 is 69.500 FCFS trial 18 is 29.800	SJF trial 1 is 40.800 SJF trial 2 is 41.900 SJF trial 3 is 46.800 SJF trial 4 is 40.800 SJF trial 5 is 39.700 SJF trial 6 is 40.200 SJF trial 7 is 41.500 SJF trial 8 is 38.900 SJF trial 9 is 45.800 SJF trial 10 is 44.400 SJF trial 11 is 40.000 SJF trial 12 is 45.700 SJF trial 13 is 52.900 SJF trial 14 is 35.900 SJF trial 15 is 20.400 SJF trial 16 is 31.900 SJF trial 17 is 50.600 SJF trial 18 is 23.700	RR2 trial 1 is 67.300 RR2 trial 2 is 70.700 RR2 trial 3 is 79.600 RR2 trial 4 is 68.100 RR2 trial 5 is 68.300 RR2 trial 6 is 69.200 RR2 trial 7 is 70.700 RR2 trial 8 is 67.600 RR2 trial 9 is 77.200 RR2 trial 10 is 73.100 RR2 trial 11 is 68.200 RR2 trial 12 is 78.300 RR2 trial 13 is 94.000 RR2 trial 14 is 60.100 RR2 trial 15 is 32.400 RR2 trial 16 is 53.500 RR2 trial 17 is 89.600 RR2 trial 18 is 37.400	RR5 trial 1 is 66.900 RR5 trial 2 is 67.400 RR5 trial 3 is 80.600 RR5 trial 4 is 66.400 RR5 trial 5 is 66.600 RR5 trial 6 is 69.400 RR5 trial 7 is 63.400 RR5 trial 8 is 71.900 RR5 trial 9 is 77.900 RR5 trial 10 is 72.600 RR5 trial 11 is 65.700 RR5 trial 12 is 82.300 RR5 trial 13 is 92.500 RR5 trial 14 is 55.500 RR5 trial 15 is 34.200 RR5 trial 16 is 56.200 RR5 trial 17 is 91.900 RR5 trial 18 is 36.200

	FCFS trial 19 is 54.200 FCFS trial 20 is 44.400  FCFS average over 20 trials is 55.290	SJF trial 19 is 32.100 SJF trial 20 is 38.700  SJF average over 20 trials is 39.635	RR2 trial 19 is 54.900 RR2 trial 20 is 65.700  RR2 average over 20 trials is 67.295	RR5 trial 19 is 55.900 RR5 trial 20 is 61.200  RR5 average over 20 trials is 66.735
15 jobs	FCFS trial 1 is 79.267 FCFS trial 2 is 79.533 FCFS trial 3 is 97.467 FCFS trial 4 is 86.800 FCFS trial 5 is 86.267 FCFS trial 6 is 107.467 FCFS trial 7 is 88.400 FCFS trial 8 is 82.800 FCFS trial 9 is 80.733 FCFS trial 10 is 90.800 FCFS trial 11 is 82.200 FCFS trial 12 is 107.133 FCFS trial 13 is 84.133 FCFS trial 14 is 70.400 FCFS trial 15 is 106.733 FCFS trial 16 is 91.733 FCFS trial 17 is 92.200 FCFS trial 18 is 80.467 FCFS trial 19 is 89.867 FCFS trial 20 is 66.267  FCFS average over 20 trials is 87.533	SJF trial 1 is 62.600 SJF trial 2 is 56.067 SJF trial 3 is 66.800 SJF trial 4 is 55.800 SJF trial 5 is 48.600 SJF trial 6 is 87.933 SJF trial 7 is 63.800 SJF trial 8 is 62.267 SJF trial 9 is 62.267 SJF trial 10 is 61.067 SJF trial 11 is 65.200 SJF trial 12 is 89.933 SJF trial 13 is 63.400 SJF trial 14 is 48.133 SJF trial 15 is 86.467 SJF trial 16 is 63.267 SJF trial 17 is 53.933 SJF trial 18 is 55.867 SJF trial 19 is 74.800 SJF trial 20 is 44.067  SJF average over 20 trials is 63.613	RR2 trial 1 is 110.600 RR2 trial 2 is 97.467 RR2 trial 3 is 117.600 RR2 trial 4 is 99.000 RR2 trial 5 is 85.667 RR2 trial 6 is 161.067 RR2 trial 7 is 110.867 RR2 trial 8 is 110.600 RR2 trial 9 is 111.000 RR2 trial 10 is 106.667 RR2 trial 11 is 113.467 RR2 trial 12 is 160.467 RR2 trial 13 is 112.333 RR2 trial 14 is 83.800 RR2 trial 15 is 154.933 RR2 trial 16 is 113.200 RR2 trial 17 is 98.133 RR2 trial 18 is 98.200 RR2 trial 19 is 129.533 RR2 trial 20 is 76.133  RR2 average over 20 trials is 112.537	RR5 trial 1 is 102.400 RR5 trial 2 is 99.333 RR5 trial 3 is 122.733 RR5 trial 4 is 99.133 RR5 trial 5 is 88.733 RR5 trial 6 is 157.933 RR5 trial 7 is 111.867 RR5 trial 8 is 108.000 RR5 trial 9 is 107.733 RR5 trial 10 is 107.067 RR5 trial 11 is 108.933 RR5 trial 12 is 157.933 RR5 trial 13 is 113.867 RR5 trial 14 is 80.267 RR5 trial 15 is 154.933 RR5 trial 16 is 114.533 RR5 trial 17 is 104.467 RR5 trial 18 is 98.867 RR5 trial 19 is 119.933 RR5 trial 20 is 70.733  RR5 average over 20 trials is 111.470

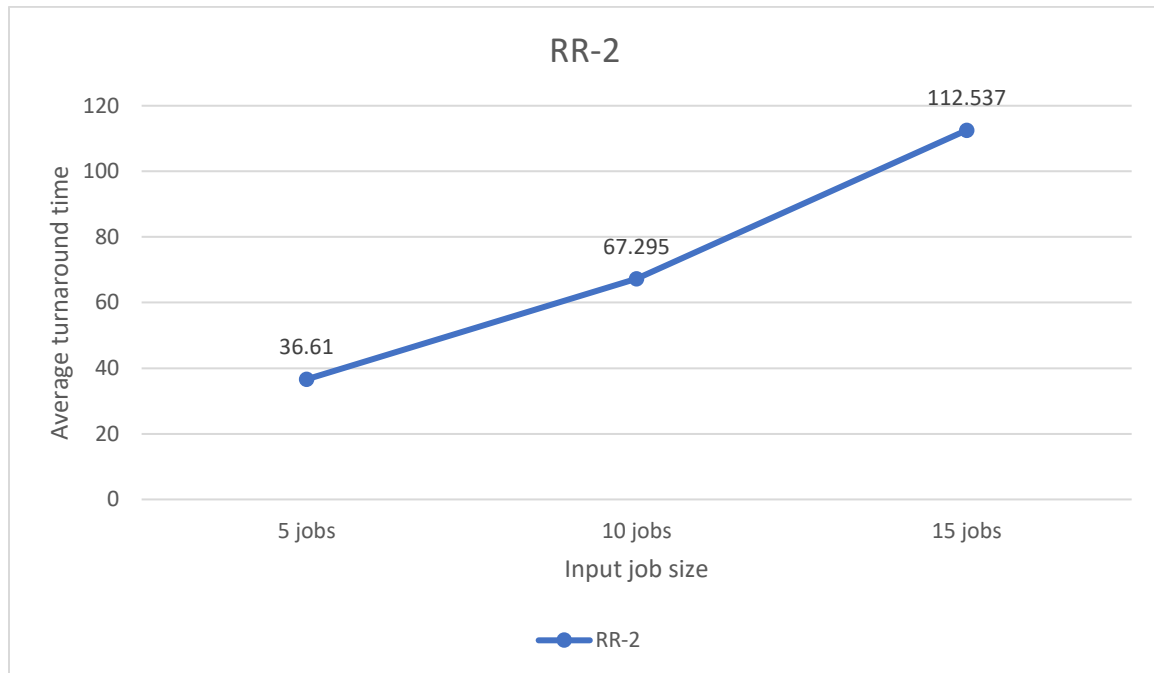
- b. Plot a graph of each algorithm, average turnaround time vs input size (# of jobs), and summarize the performance of each algorithm based on its own graph.



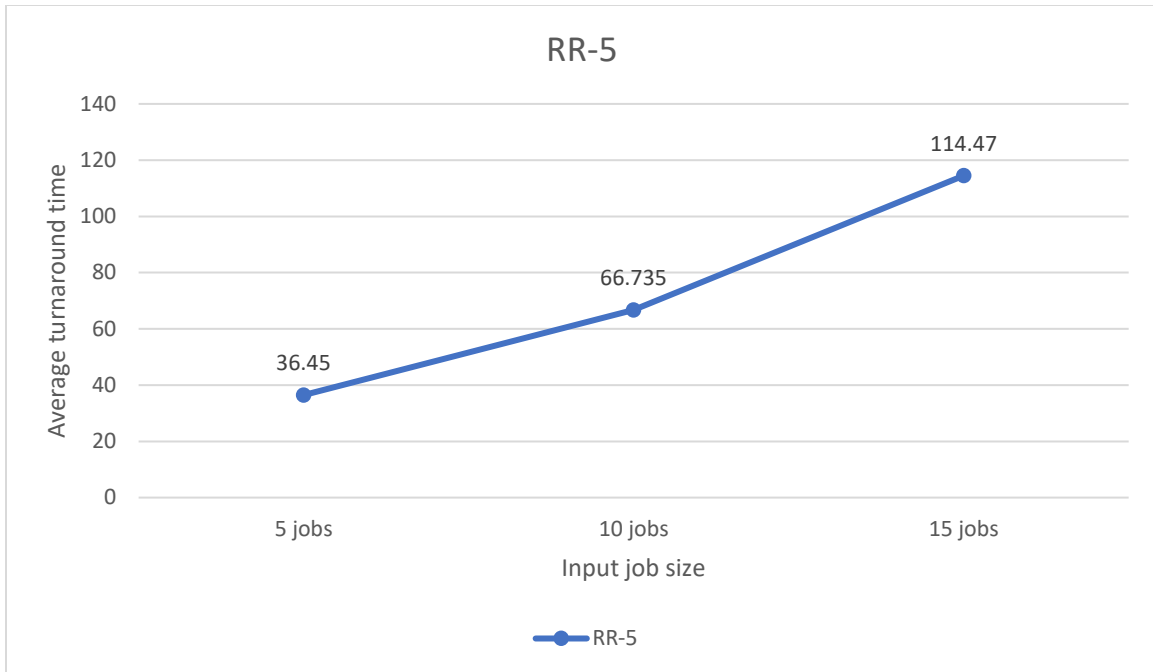
The first come first serve is a non-preemptive algorithm and the easiest algorithm to implement in code. I used the Java library linked list for the data structures which according to the Java website it includes the function as stack and queue. The average turnaround time increase when the job size increase, so it's not the most efficient algorithm for the scheduling algorithm. Because the turnaround time is not minimized.



The shortest job first algorithm is not preemptive and it picks the shortest job first. This algorithm is not difficult to implement too. I just need to add a custom comparator into my link list data structure so it will automatically sort the shortest job first. The shortest job first algorithm got the best average turnaround time to compare to the other 3 algorithms, so it probably the best way for CPU to scheduling jobs.

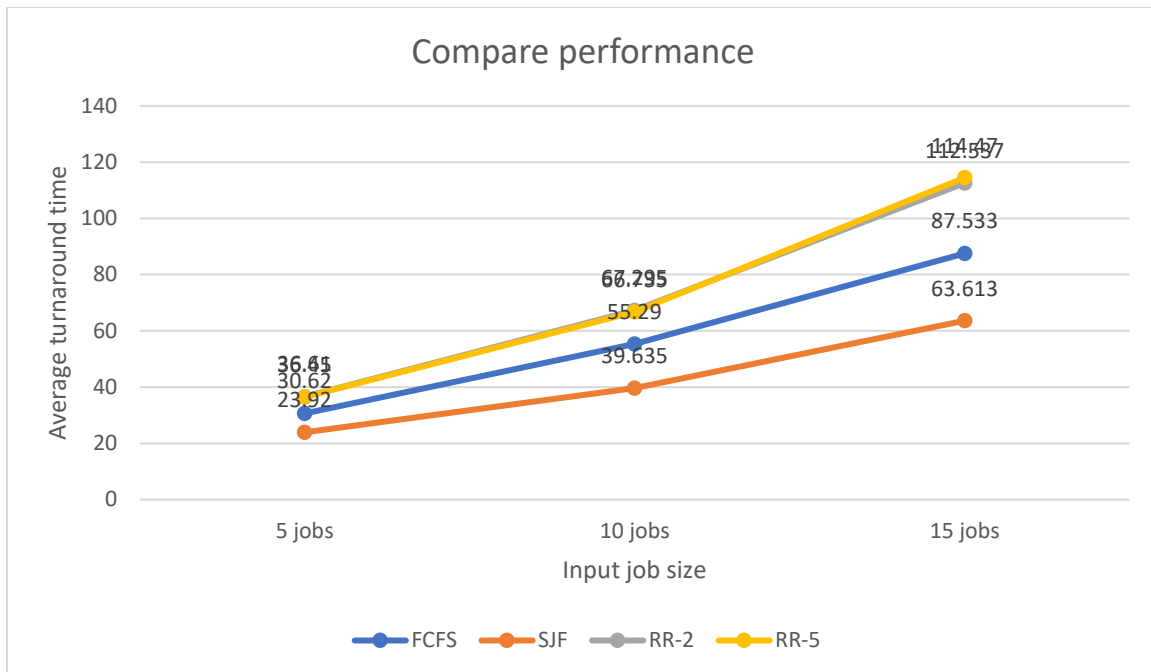


The round-robin pretty complex algorithm to write compare to the other 3, but it's surprising to see the performance is the worst amount the three. At 15 jobs the average turnaround time is almost double the SJF average turnaround time, so I felt this algorithm is kind of pointless.



The round-robin with time slices 5 is almost the same compare to slices 2. I thought it will get a better average turn around time but it didn't. The process of switching around the jobs wasted lots of time to calculate. I wanted to increase the slices to 10 in my free time and see if the average turnaround time will decrease or not.

Plot all four graphs on the same graph and compare the performance of all four algorithms. Rank four scheduling algorithms. Try giving the reasons for the findings.



From the above graph, we can see the round-robin 2 and 5 slices almost have no difference in their average turnaround time. The shortest job first performs the best compare to the other 3 algorithms, and it's the simplest algorithm to write too. This experiment is very helpful for us to see the difference between each of the algorithms, and also a good project to practice what we learn from the data structure class.

- c. Conclude your report with the strength and constraints of your work. At least 100 words. (Note: It is reflection of this project. If you have a change to re-do this project again, what you like to keep and what you like to do differently in order get a better quality of results.)

I think this project is pretty simple and fun to help us refresh what we had learned from the data structure class. I only spend about 1 day to finish the first part of the implementation, and another day to make the testing strategy for the programs. If I have the chance to re-do this project again I would like to try different data structures for this project such as hashtable or priority queue. I want to be familiar with each of the algorithms that we have learned before, so in the future, it will be easy for me to use those algorithms for work. Also, I would like to try with this with different coding languages such as C++, C#, or python so I can understand the between those languages.