DEPARTMENT OF SOFTWARE ENGINEERING

The University of Lahore

Faculty of Information Technology

Assignment Cover Letter

(Individual Work)

Student Name	Atta Elahi	Program	BS(SE)
SAP ID	70082385	Title of Assignment	02
Course Code	CS 11303	Due Date	16/10/2022
Course Name	Operating Systems	Submission Date	07/11/2022
Section	T		

The assignment should meet the below requirements:

1- Assignment (hard copy) is required to be submitted on clean paper and soft copy as per lecturer's instructions.
2- Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3- The above information is complete and legible.
4- Compiles pages are firmly attached.
5- Assignment has been copied (softcopy & hardcopy) for each student ahead of the submission.

Plagiarism/Cheating

The university seriously regards all forms of plagiarism, cheating and collision as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the University.

Declaration of Originality

By signing this assignment, I understand, accept and consent to The University of Lahore terms and policies on plagiarism.

I hereby declare that this work represents my own effort, and that all text and code have been written by me and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature	

Question 1: Round robin schedulers normally maintain a list of all runnable processes, with each process occurring exactly once in the list. What would happen if a process occurred twice in the list? Can you think of any reason for allowing this?

If a processor occurs more than once in the round-robin list, then it will get two turns for each pass through the list. One reason for allowing this would be to implement a primitive priority system since the more times it occurs on the list, the higher the percentage of time the CPU will spend on that process.

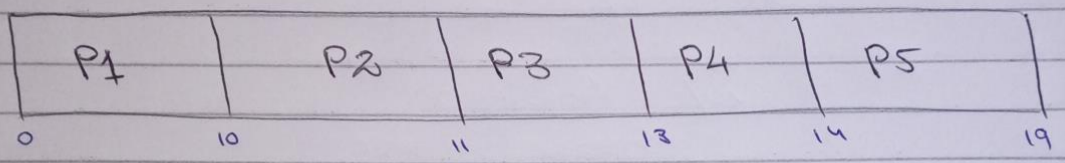
Question 2: Most round robin schedulers use a fixed size quantum. Give an argument in favor of a small quantum. Now give an argument in favor of a large quantum.

An argument against a small time quantum: Efficiency. A small time quantum requires the timer to generate interrupts with short intervals. Each interrupt causes a context switch, so overhead increases with a larger number of interrupts. An argument for a small time quantum: Response time. A large time quantum will reduce the overhead of context switching since interrupts will be generated with relatively long intervals, hence there will be fewer interrupts. However, a short job will have to wait longer time on the ready queue before it can get to execute on the processor. With a short time quantum, such a short job will finish quicker and produces the result to the end user faster than with a longer time quantum.

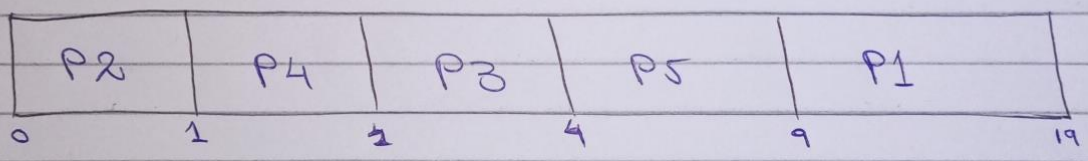
Question 3: Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

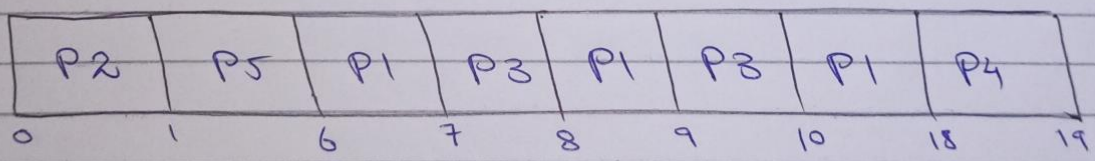
a: FCFS:



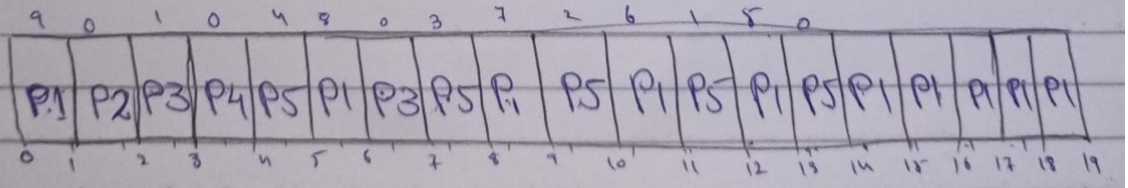
SJF: Shortest Job First



Non-Preemptive Priority:



RR: Round Roubin:



B: Tornaround -time of each process

	FCFS	SJF	NPP	RR
P1	0+10 = 10ms	9+10 = 19ms	18ms	19ms
P2	10+1 = 11ms	0+1 = 1ms	1ms	2ms
P3	11+2 = 13ms	2+2 = 4ms	10ms	7ms
P4	13+1 = 14ms	1+1 = 2ms	19ms	4ms
P5	14+5 = 19ms	4+5 = 9ms	6ms	14ms

C: Waiting Time:

	FCFS	SJF	NPP	RR
P1	0ms	9ms	8ms	9ms
P2	10ms	0ms	0ms	1ms
P3	11ms	2ms	8ms	5ms
P4	13ms	1ms	18ms	3ms
P5	14ms	4ms	1ms	9ms

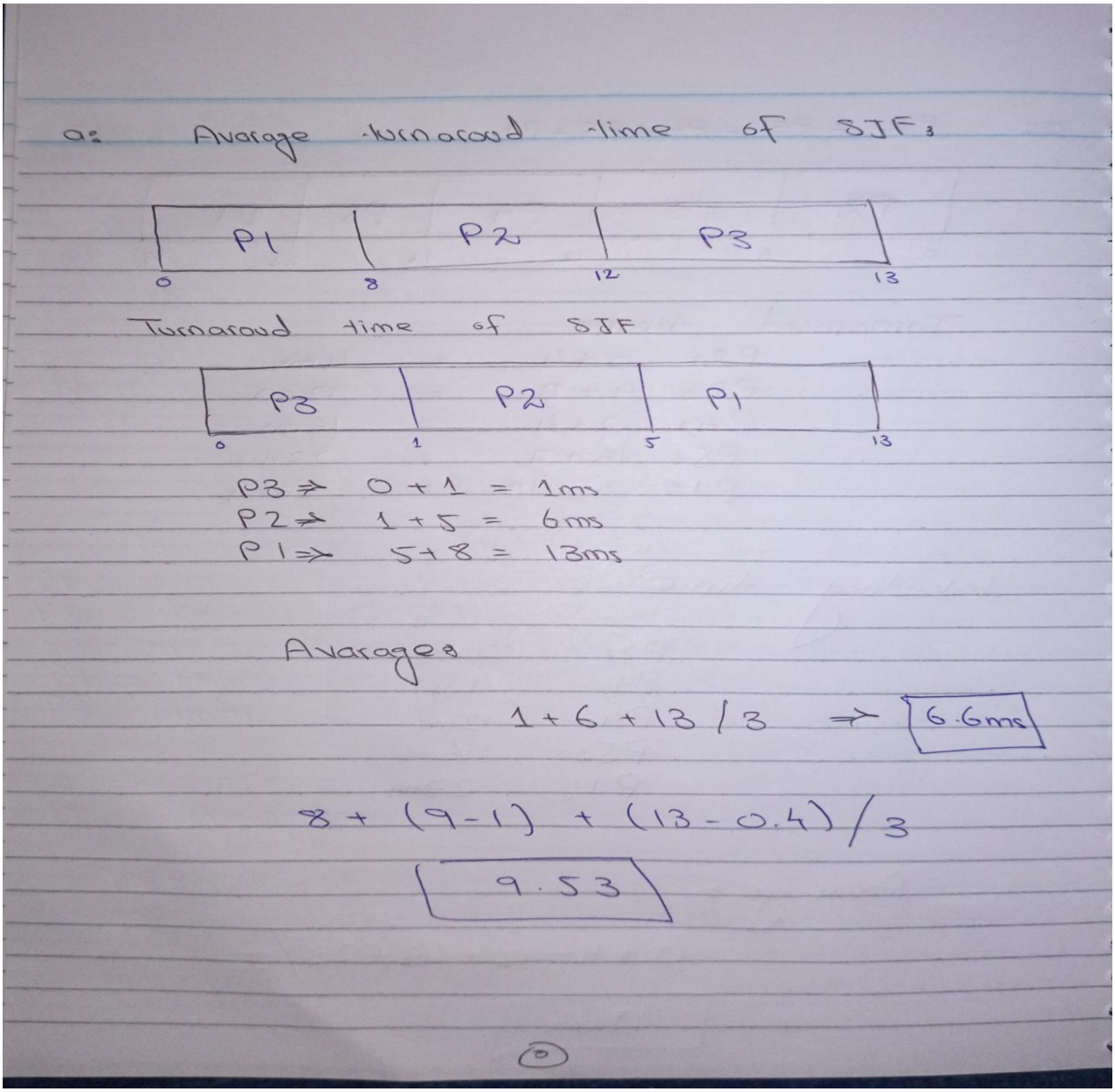
D: Avarage:

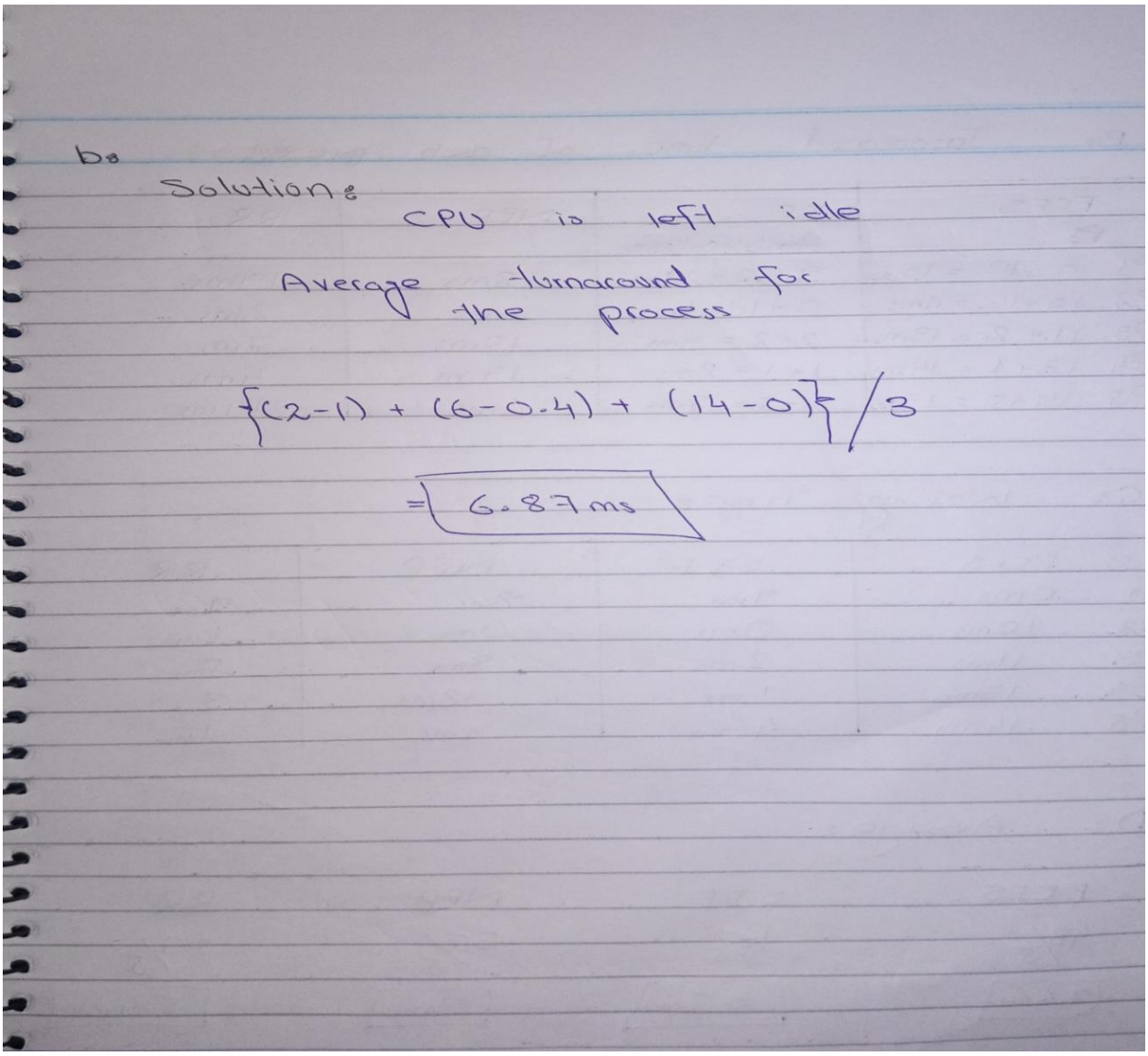
FCFS	SJF	NPP	RR
48/5	16/5	35/5	27/5
9.6ms	3.2ms	7ms	5.4ms

SJF has minimal avarage.

Question 4: Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of time. In answering the questions, use non-preemptive scheduling and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P_1	0.0	8
P_2	0.4	4
P_3	1.0	1

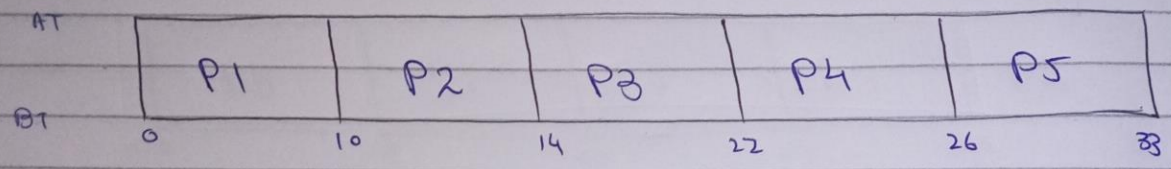




Question 5: Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst
P1	10
P2	4
P3	8
P4	4
P5	7

a: FCFS Grand chart



b: Turnaroud time

P1: 0 + 10 = 10ms
P2: 10 + 4 = 14ms
P3: 14 + 8 = 22ms
P4: 22 + 4 = 26ms
P5: 26 + 7 = 33ms

Waiting time:

P1: 0ms
P2: 10ms
P3: 14ms
P4: 22ms
P5: 26ms

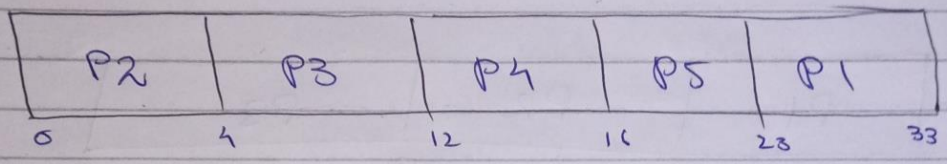
Avarage:

0 + 10 + 14 + 22 + 26 / 5

14.4ms

(5)

C: P2 < P3 < P4 < P5 < P1



Turnaround time:

P2: 0+4 = 4ms
P3: 4+8 = 12ms
P4: 12+4 = 16ms
P5: 16+7 = 23ms
P1: 23+10 = 33ms

Waiting time

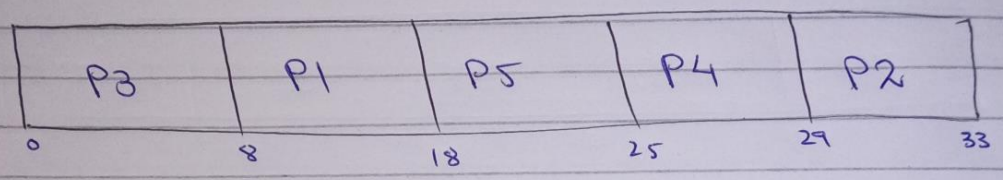
P2: 0ms
P3: 4ms
P4: 12ms
P5: 16ms
P1: 23ms

Avarage:

0+4+12+16+23/5

11ms

d: P3 < P1 < P5 < P4 < P2



Turnaround time:

P3:	0+8	=	8ms
P1:	8+10	=	18ms
P5:	18+7	=	25ms
P4:	25+4	=	29ms
P2:	29+4	=	33ms

Waiting time:

P3:	0
P1:	8
P5:	18
P4:	25
P2:	29

Average:

$0 + 8 + 18 + 25 + 29 / 5$

16ms