1.

Friday, February 16, 2024 7:01 PM

Problem 1

Consider the following processes:

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3

Show how the above processes execute over time on a single CPU system. Compute the completion time for each process, the average turnaround time and the average normalized turnaround time for all processes under each of the following schedulers:

(a) FCFS. **[5 pts]**

(b) Round Robin with (q = 1). **[5 pts]**

(c) Round Robin with (q = 2). **[5 pts]**

(d) Shortest Job First. **[5 pts]**

(e) Shortest Remaining Time First. **[5 pts]**

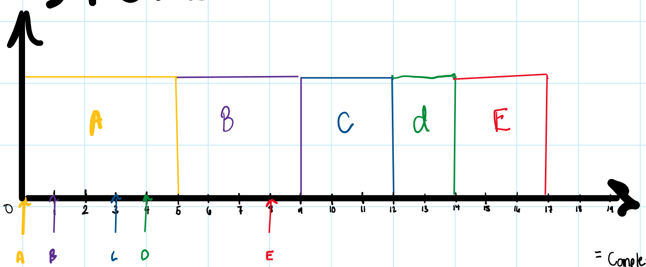
(f) HRRN. **[5 pts]**

(g) Multi-level Feedback with 2 queues. Queue serves 1 quantum (unit of time) at a time while queue 2 serves 2 quantums at a time. All processes get serviced from queue 1 initially in FCFS fashion and if they do not complete, they move to queue 2. Queue 2 runs a round robin scheduler. Assume that Queue 1 has a higher priority than Queue 2, and assume that a process arriving to queue 1 cannot preempt an already running process from queue 2 *within* its 2 quantum. **[5 pts]**

(h) PHRRN is a new scheduler proposed by some students based on their discussions of the regular HRRN. PHRRN stands for preemptive highest response ratio next. In PHRRN, the scheduler runs periodically (in addition to when a process completes) and the ratios are computed at that time to make scheduling decisions. Show the execution of the above processes when the scheduler is invoked every 2 time units. **[5 pts]**

call scheduler compute ratio1

a) FCFS

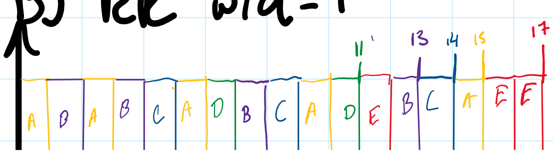


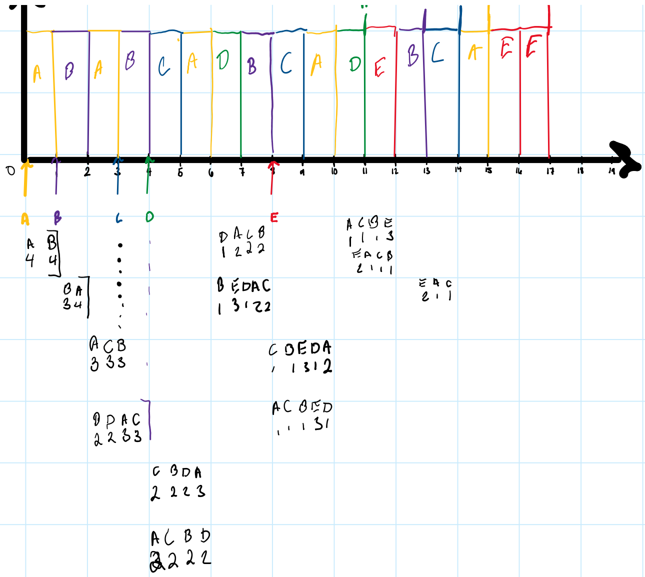
= Complete − Arrival

= $\frac{\text{TurnAround}}{\text{Service}}$

	Arrival Time	Service Time	Completion Time	TurnAround	AVG NORM TurnAround
A	0	5	5	5	5/5 = 1
B	1	4	9	8	8/4 = 2
C	3	3	12	9	9/3 = 3
D	4	2	14	10	10/2 = 5
E	8	3	17	9	9/3 = 3
AVG				8.2	2.8

B) RR w/ q = 1



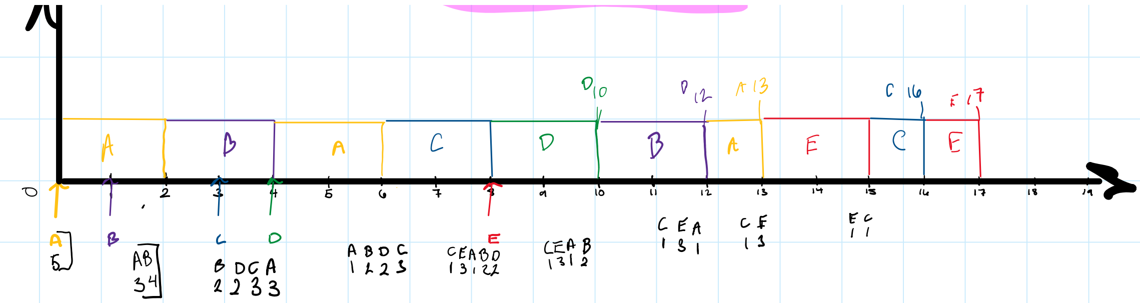


	Arrival Time	Service Time	Completion Time	= Complete − Arrival Turn Around	= Turn Around / Service AVG NRM Turn Around
A	0	5	15	15	15/5 = 3
B	1	4	13	12	12/4 = 3
C	3	3	14	11	11/3 = 3.6
D	4	2	11	7	7/2 = 3.5
E	8	3	17	9	9/3 = 3
AVG				8.8	3.22

C) RR q=2

· what happens when process finishes before q

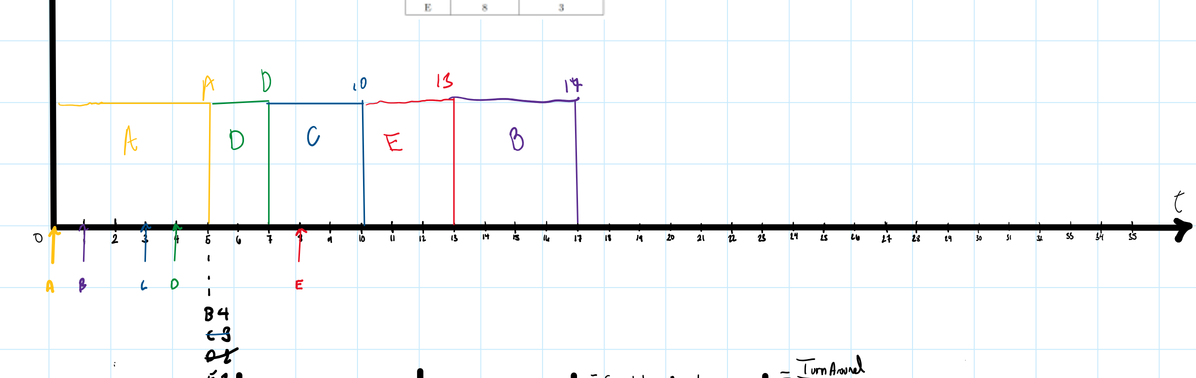
D 10 B 12 A 13 C 16 E 17



	Arrival Time	Service Time	Completion Time	= Complete − Arrival TurnAround	$= \frac{\text{TurnAround}}{\text{Service}}$ AVG NRM TurnAround
A	0	5	13	13	13/5
B	1	4	12	11	11/4
C	3	3	16	13	13/3
D	4	2	10	6	6/2 = 3
E	8	3	17	9	9/3 = 3
AVG				10.4	3.137

D) Shortest Job First

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3

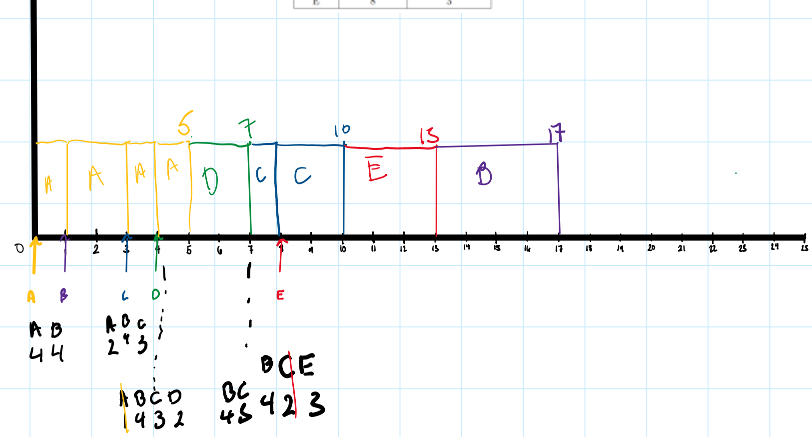


	Arrival Time	Service Time	Completion Time	= Complete − Arrival TurnAround	$= \frac{\text{TurnAround}}{\text{Service}}$ AVG NRM TurnAround
A	0	5	5	5	5/5 = 1

	Arrival Time	Service Time	Completion Time	Turn Around	AVG NRM Turn Around
A	0	5	5	5	5/5 = 1
B	1	4	17	16	16/4 = 4
C	3	3	10	7	7/3
D	4	2	7	3	3/2
E	8	3	13	5	5/3
AVG				7.2	2.1

E) Shortest Remaining Time First

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3



	Arrival Time	Service Time	Completion Time	= Complete − Arrival Turn Around	= $\frac{\text{Turn Around}}{\text{Service}}$ AVG NRM Turn Around
A	0	5	5	5	5/5 = 1
B	1	4	17	16	$\frac{16}{4}$ = 4
C	3	3	10	7	7/3

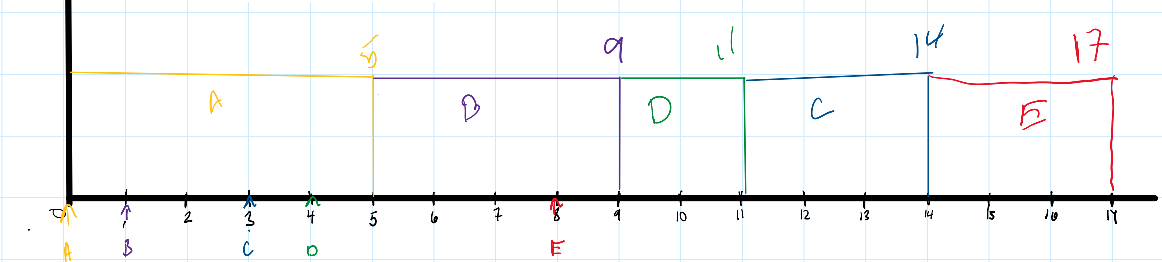
C	3	3	10	7	7/3
D	4	2	7	3	3/2
E	8	3	13	5	5/3
AVG				7.2	2.1

F) Highest Remaining Ratio Next

$$R = \frac{W+S}{S}$$

w : wait t

s : service t



Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3

@5 $R_B = \frac{4+4}{4} = 2$

$R_C = \frac{2+3}{3} = \frac{5}{3} = 1.6$

$R_D = \frac{1+2}{2} = 1.5$

@9 $R_C = \frac{6+3}{3} = 3$

$R_D = \frac{5+2}{2} = \frac{7}{2} = 3.5$

$R_E = \frac{1+3}{3} = 1.3$

@11 $R_C = \frac{8+3}{3} = 3.67$

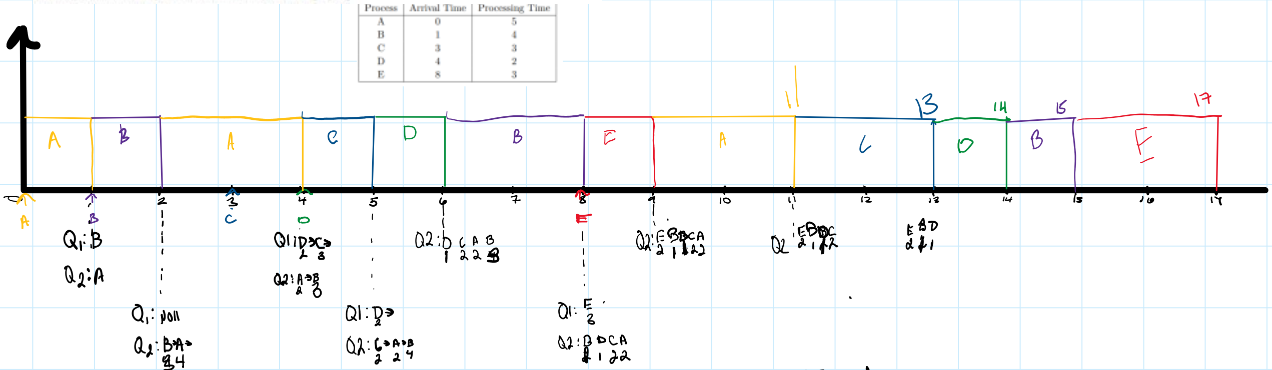
$R_E = \frac{3+3}{3} = 6..$

	Arrival Time	Service Time	Completion Time	= Complete − Arrival Turn Around	= Turn Around / Service AVG NRM Turn Around
A	0	5	5	5	5/5 = 1
B	1	4	9	8	8/4 = 2
C	3	3	14	11	11/3 = 3.667

C	3	3	14	11	$\frac{11}{3} = 3.667$
D	4	2	11	7	$\frac{7}{2} = 3.5$
E	8	3	17	9	$\frac{9}{3} = 3$
AVG				8	2.633

(g) Multi-level Feedback with 2 queues. Queue serves 1 quantum (unit of time) at a time while queue 2 serves 2 quantums at a time. All processes get serviced from queue 1 initially in FCFS fashion and if they do not complete, they move to queue 2. Queue 2 runs a round robin scheduler. Assume that Queue 1 has a higher priority than Queue 2, and assume that a process arriving to queue 1 cannot preempt an already running process from queue 2 *within* its 2 quantum. **[5 pts]**

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3



	Arrival Time	Service Time	Completion Time	= Complete − Arrival Turn Around	= Turn Around / Service AVG NORM Turn Around
A	0	5	11	11	$11/5$
B	1	4	15	14	$\frac{14}{4} = \frac{7}{2}$
C	3	3	13	10	$\frac{10}{3}$
D	4	2	14	10	$\frac{10}{2} = 5$
E	8	3	17	9	$\frac{9}{3} = 3$
				14	2.407

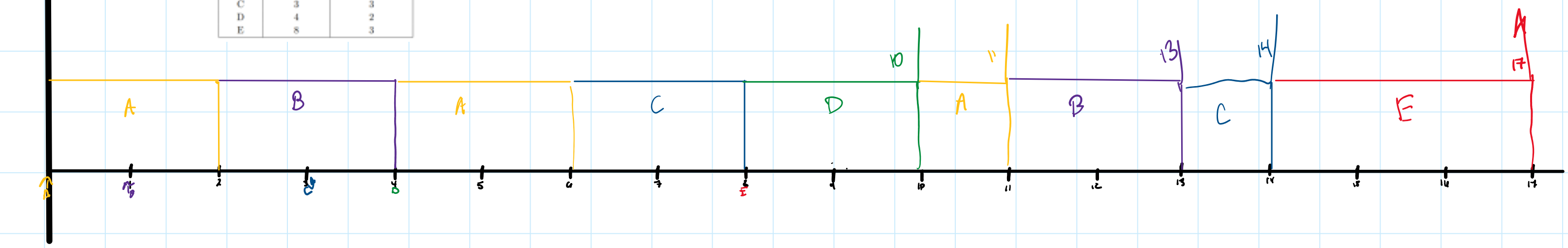
E	8	3			
AVG				14	3.407

(h) PHRRN is a new scheduler proposed by some students based on their discussions of the regular HRRN. PHRRN stands for preemptive highest response ratio next. In PHRRN, the scheduler runs periodically (in addition to when a process completes) and the ratios are computed at that time to make scheduling decisions. Show the execution of the above processes when the scheduler is invoked every 2 time units. **[5 pts]**

Process	Arrival Time	Processing Time
A	0	5
B	1	4
C	3	3
D	4	2
E	8	3

S = REMAINING SERVICE TIME

W = TIME WAITING SINCE Arrival/Preempt



@2 $R_a = \frac{0+3}{3} = 1$

$R_b = \frac{1+4}{4} \frac{5}{4}$

@4 $R_a = \frac{2+3}{3} = 5/3$

$R_b = \frac{0+2}{2} = 1$

$R_c = \frac{1+3}{3} = \frac{4}{3}$

$R_d = \frac{0+2}{2} = 1$

@6) $R_a = \frac{0+1}{1} = 1$

$R_B = \frac{2+2}{2} = 2$

$R_c = \frac{3+3}{3} = 2$

$R_d = \frac{2+2}{2} = 2$

@8) $R_a = \frac{2+1}{1} = 3$

$R_B = \frac{4+2}{2} = 3$

$R_C = \frac{0+1}{1} = 1$

$R_d = \frac{4+2}{2} = 3$

$R_E = \frac{0+3}{3} = 1$

@10) $R_a = \frac{4+1}{1} = 5$

$R_B = \frac{6+2}{2} = 4$

$R_c = \frac{2+1}{1} = 3$

$R_E = \frac{2+3}{3} = 5/3$

@11) $R_B = \frac{7+2}{2} = \frac{9}{2} = 4\frac{1}{2}$

$R_c = \frac{3+1}{1} = 4$

$R_E = \frac{3+3}{3} = 2$

@13) $R_c = \frac{5+1}{1} = 6$

$R_E = \frac{5+3}{3} = 8/3$

2.

Friday, February 16, 2024 7:06 PM

Problem 2

Prove that the Shortest Job First scheduling algorithm achieves the minimum average waiting time for a group of processes that arrived at the same time.[**5 pts**]

n processes $i=1,\ldots,n$

X_i: Service time requested by process i

maybe there's another that does it better*
proof by contradiction

SJF scheduling algorithm achieves the minimum avg wait time for a group of processes that arrive at the same time

Consider n process that arrive at the same time where X_i is the service time requested by process i & $i=1,2,\ldots,n$, let $n=3$

Assume that SJF DOESN'T achieve the minimum AVG waiting time, let's find the AVG wait time for other orders & service times $X_i = i \Rightarrow X_1 = 1$ sec

AlgX: i_3, i_2, i_1 W_{i_1}, wait time for i_1

$W_{i_3} = 0,\ W_{i_2} = X_3,\ W_{i_1} = X_3 + X_2$

$= 0,\ 3,\ 5$

AVG = 8/3

$AVG_x = 8/3$

$AlG_y = i_2, i_3, i_1$

$w_2 = 0, \quad w_{i3} = x_2, \quad w_{i_1} = x_2 + x_3$

$= 0 \qquad = 2 \qquad = 4+3$

$AVG_y = 7/3$

$AlG_z = i_1, i_3, i_2$

$w_1 = 0, \quad w_2 = x_3 + x_1, \quad w_3 = x_1$

$0 + 4 + 1 = 5$

$AVG_Z = 5/3$

$AVG_{SJF} > AVG_x \,||\, AVG_y \,||\, AVG_Z$

$AlG_{SJF} = i_1, i_2, i_3$

$w_1 = 0, \quad w_2 = x_1, \quad w_3 = x_1 + x_2$

$0 + 1 + 3 = 4$

$AVG_{SJF} = 4/3 \leq (AVG_x)(AVG_y)(AVG_z)$

Which contradicts the assumption SJF doesn't achieve the minimum AVG wait time B/C it does

3.

Friday, February 16, 2024 7:06 PM

Problem 3

Least Slack Process Next (LSPN) is a real-time scheduler for periodic tasks. Slack is the amount of time between when a task would complete if it started now and its next deadline. Thus it can be expressed as:

$$Slack = D - t - C \qquad (1)$$

where D is the deadline time, t is the current time and C is the processor time needed. LSPN selects the task with the minimum slack time to execute next. If two tasks have the same slack, they are serviced based on FCFS. Answer the following questions:

(a) What does it mean for a task to have a slack of 0? **[2 pts]**

$(D-t)-C$

$(D-t)$ is the ABSOLUTE MINIMUM amount of time for processor to finish task. Thus, if Slack=0 $(D-t)-C=0$, NO SLACK CPU MUST START NOW

(b) What does it mean for a task to have a negative slack? **[2 pts]**

It means its impossible for CPU to have enough time to finish task

(c) How long may the scheduler delay starting a task (and still meet its deadline), if that task has a slack s? **[2 pts]**

Scheduler may delay s timeunits

DUH!?

(d) Consider 3 period tasks: A, B and C. Task A has a period 6 and execution time 2, task B has a period of 8 and execution time of 2 and task C has a period of 12 and execution time of 3. Illustrate (by drawing the executions of A, B and C over time) how LSPN would schedule these tasks in comparison to Earliest Deadline First and Rate Monotonic Scheduling. **[9 pts]**

TASK	Period	EXE
A	6	2
B	8	2
C	12	3

@0 $S_A = 6-0-2 = 4$
$S_B = 8-0-2 = 6$
$S_C = 12-0-3 = 9$

@2 $S_A = 12-2-2 = 8$
$S_B = 8-2-2 = 4$
$S_C = 12-2-3 = 7$

@4 $S_A = 12-4-2 = 6$
$S_B = 16-4-2 = 10$
$S_C = 12-4-3 = 5$

@7 $S_A = 12-7-2 = 7$
$S_B = 16-7-2 = 7$
$S_C = 24-7-3 = 4$

@9 $S_A = 18-9-2 = 7$
$S_B = 16-9-2 = 5$
$S_C = 2$

@11 $S_A = 18-11-2 = 5$
$S_B = 24-11-2 = 11$
$S_C = 24-11-3 = 10$

@13 $S_A = 24-13-2 = 9$
$S_B = 24-13-2 = 9$
$S_C = 24-13-3 = 8$

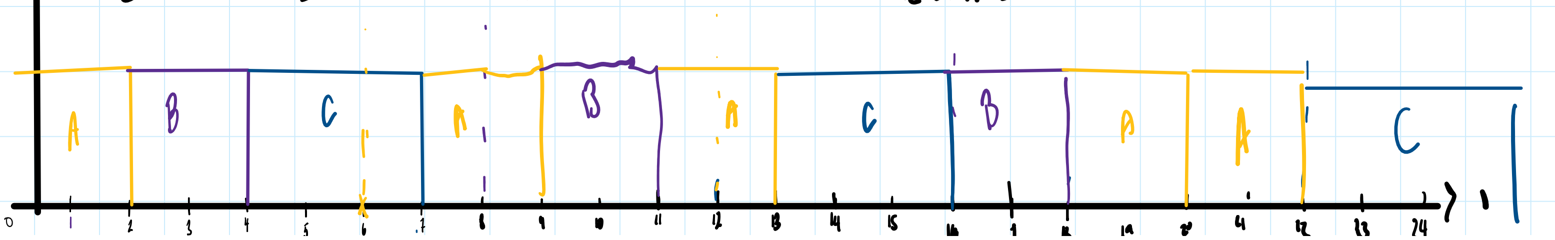
@16 $S_A = 24-16-2 = 6$
$S_B = 6$
$S_C = 36-16-3$

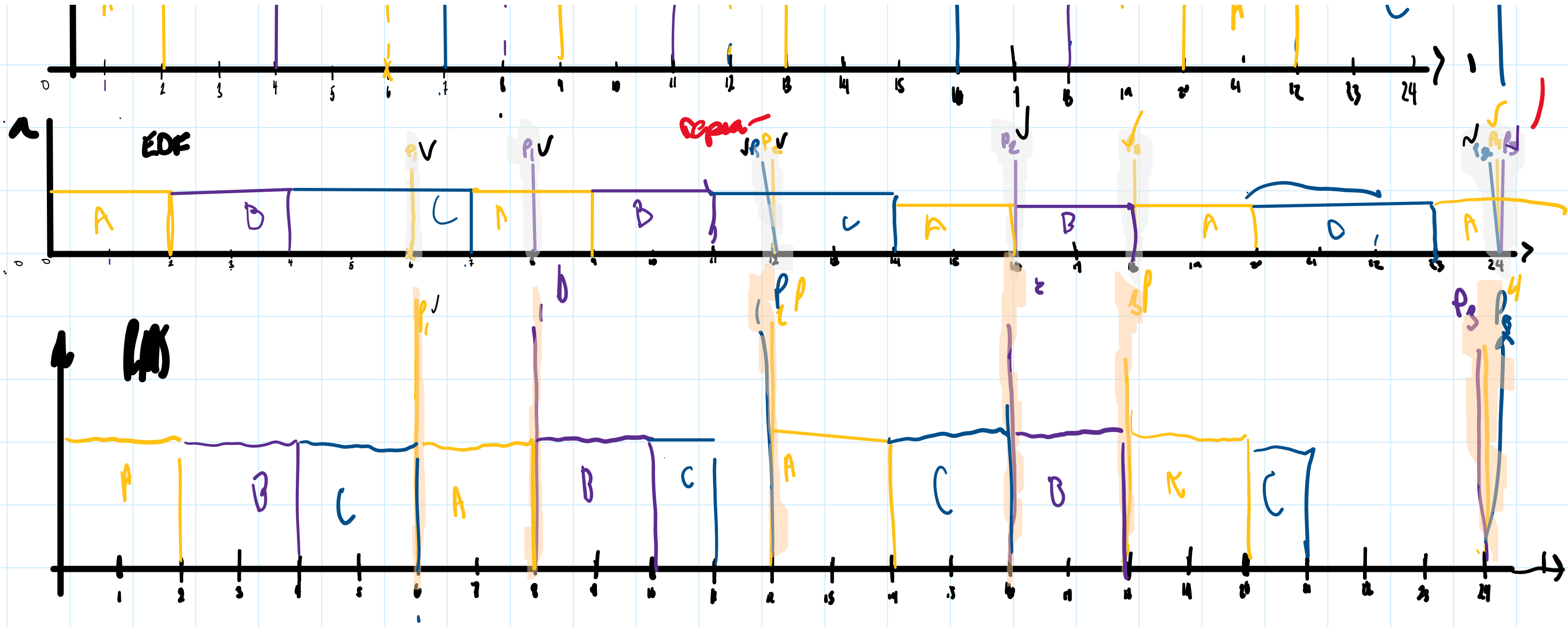
@18 $S_A = 24-18-2 = 4$
$S_B = 36-18-2 = 16$

@20 $S_A = 30-20-2 = 8$
$S_B = 36-20-2 = 14$
13

@22 $S_A = 36-22-2 = 12$
$S_B = 36-22-2 = 12$
$S_C = 36-22-3$

LEAST SLACK P NEXT





$\frac{2}{6} + \frac{2}{8} + \frac{3}{12}$

C:1

$\frac{8}{24} + \frac{6}{24} + \frac{6}{24} = \checkmark$