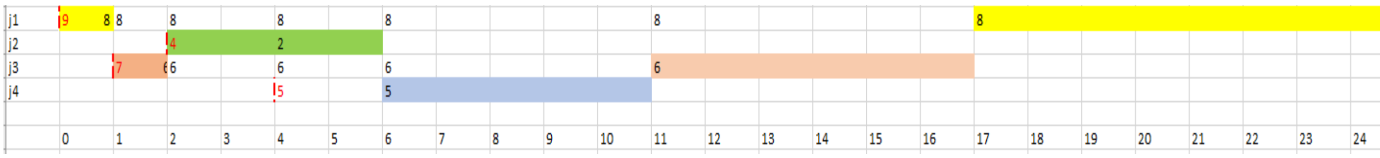
1a)SRT (shortest run time)



b)HSN(highest slowdown next)

time 9

j2= 9+4-2/4 = 2.75

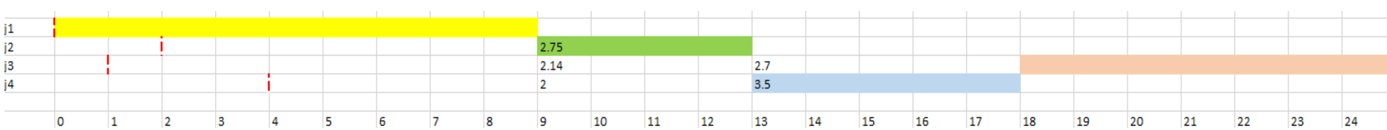
j3= 9+7-1/7 = 2.14

j4= 9+5-4/5 = 2

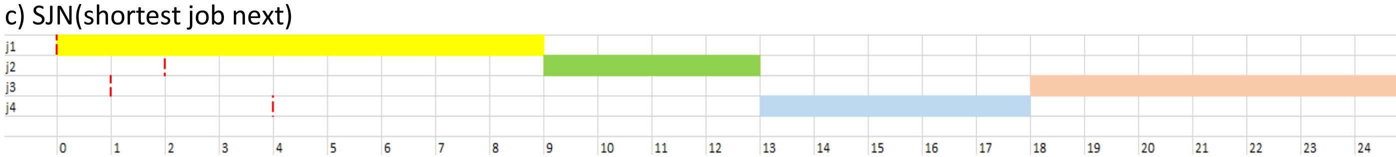
time 13

j3= 13+7-1/7=2.7

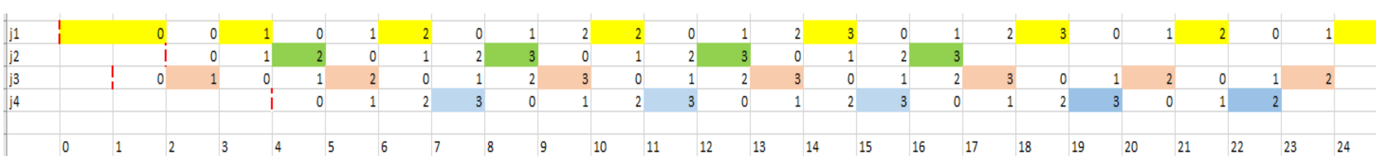
j4=13+5-4/4 = 3.5



c) SJN(shortest job next)



d)round robin



e)

SRT

Jobs	Response time
Job1	25-0=25
Job2	6-2= 4
Job3	17-1= 16
Job4	11-4 = 7
Average	(25+4+16+7)/4 = 13

HSN

Jobs	Response time
Job1	9-0=9
Job2	13-2 = 11
Job3	25-1= 24
Job4	18-4 = 14
Average	(9+11+24+14)/4 = 14.5

SJN

Jobs	Response time
Job1	9-0=9
Job2	13-2 = 11
Job3	25-1= 24
Job4	18-4 = 14
Average	(9+11+24+14)/4 = 14.5

RR

Jobs	Response time
Job1	25-0= 25
Job2	17-2 = 15
Job3	24-1=23
Job4	23-4=19
Average	(25+15+23+19)/4 = 20.5

Based on the calculation we can say that the best average response time performance is SRT. It will have the best average response since a job does not have to reach completion before starting the next job. Hence, we can have shorter job finishing first, which meant those jobs will have a shorter response time, hence over all SRT should have a better average response time.

e)

Jobs	Slow down on job2 (response time /job length)
SRT	4/4= 1
HSN	11/4 = 2.75
SJN	11/4= 2.75
RR	15/4= 3.75

The best in terms of fairness is SRT.

We can look at the job with the shortest length. We compare the slowdown, and the fairest one will be the one that allocate the lowest slowdown to the job with the shortest one. SRT will be fair since it will puts a longer job (more remaining time) on hold as it process the shorter job first .Hence, the respsonse time for a shorter job will be lower in a SRT which is unlike HSN and SJN where they have might have to wait for other longer job to finish (look at problem 3 e, it's a case where a shorter job would have to wait a long time to finish). And a shorter job will have a shorter response time than the round robin where everyone gets a relatively equal priority so shorter job will have to wait longer to complete unlike SRT.

2.

Problem 2

a) 53900

b) $10000 = z \pm \frac{\sigma}{\sqrt{n}}$

stdev = 33033.48

$10000 = z \times \frac{33033.48}{\sqrt{10}}$

$z = 0.957$

$z = 0.96 \Rightarrow 0.8315$

one tail $= (1 - 0.8315)$

two tail $= 2(1 - 0.8315)$

$1 - 2(1 - 0.8315) = 0.663$

$\boxed{66.3\%}$

c) $a = 0.05 \quad a/2 = 0.025$

$1 - \frac{a}{2} = 1 - 0.025 = 0.9750$

$z = 1.96$

$10{,}000 = 1.96 \times \frac{\sigma}{\sqrt{n}} \leftarrow \sigma = 33033.48$

$0.154451 = \frac{1}{\sqrt{n}}$

$\frac{0.154451 (\sqrt{n})}{0.154451}^2 = \left(\frac{1}{0.154451}\right)^2$

$n = 41.92 \approx 42$

$42 - 10 = 32$

$\boxed{32 \text{ more sample}}$

d) $\text{Cor}(A,B) = \frac{1}{n} \sum_{i=1}^{n} \frac{(A_i - \bar{A}) \cdot (B_i - \bar{B})}{\sigma_A \cdot \sigma_B} = \frac{\text{cov}(A,B)}{\sigma_A \sigma_B}$

$\sigma_A = 33033.45 \quad \sigma_B = 2.780887$

$\text{Cov}(AB) = 71422.22$

$\frac{71422.22}{33031.48 (2.780887)} = 0.77491$

$\boxed{\text{Since it's close to 1 it's correlated}}$

e) exclude data less than 5 yrs

$10750 = z \times \frac{33661.13}{\sqrt{8}}$

$\frac{10750}{11901} = z \times \frac{11901}{11901}$

$0.90 = z \rightarrow 0.8159$

one tail $= 1 - 0.8159$

two tail $= 2(1 - 0.8159)$

$1 - 2(1 - 0.8159) = 0.6318$

$\boxed{63.18\%}$

3a)

Job id	Job length	Arrival time
J1	3,1	0,10
J2	5	1
J3	2,4	3,11
J4	6	5
J5	2	7

b) The algorithm is shortest remaining time (SRT). We can see that at time 2, j2 arrives but we till choose to process j1 because j1 has a remaining time of 2 while j2 has a remaining time of 5. Similarly, a similar pattern is seen when j3 arrives at 3. We start j3 because it has a remaining time of 2 when compared to j2 which has a remaining time of 5. This pattern continues when choosing when to choose the next job hence, we can say that it's a srt algorithm.

c)At job 11, we have j2 with remaining time of 2, and j3 with remaining time of 4, and j4 with remaining time of 6. Hence we choose to start the one with shortest remaining time, which is j2.

d) slowdown – $t+c_i-ai/c_i$

time 3:

j2 : 3+5-1/5 = 1.4

j3: 3+2-3/2 = 1

time 5

j3 = 8+2-3/2 = 3.5

j4 = 8+6-5/6 = 1.5

j5= 8+2-7/6 = 0.5

time 10

j1 = 10+1-10/1 =1

j4 = 10+6-5/6 = 1.8

j5 = 10+2-7/2 =2.5

time 12

j1= 12+1-10/1 = 3

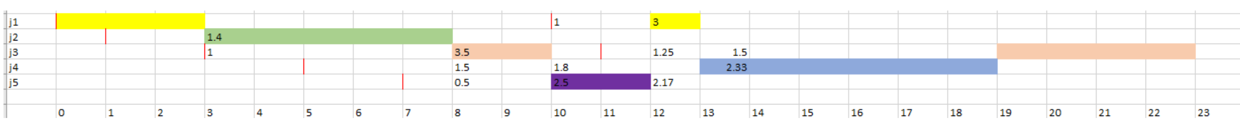
j3= 12+4-11/4 = 5/4=1.25

j4= 12+6-5/6 = 2.167

time 13

j3= 13+4-11/4 = 6/4 = 1.5

j4 = 13+6-5/6 = 14/6= 2.33



e) Assume you have a job 1 that arrives at 0, with a job length of 100 time unit while job 2 arrives at time 1 with a job length of 1 unit. In this case, since at 0 job 1 arrives and is the only job in the system so it must go to completion before we can consider job 2. This means that for job2, the slowdown will be 100-1/1 which is a slow down of 99. This will be unfair since we want to allocate the lowest slowdown to the job of the shortest length.