Problem 1.

a). State-aware, because the order of going to which station is matter, in other words, changing the order will affects the distance in between.

b). Since waiting time doesn't matter in this case and our goal is to minimize the number of direction changing, we would choose scan

c). Since the training is going toward BU, we would keep the same direction

14 -> 13 -> 6 -> 2 -> 0 -> 15 -> 16 -> 17 -> 19

d).

The algorithm I got before we learnt CAP:

We can combine SSF and the slide windows. Since there is no more than three request forehead allow, we can create a size 3 slide window, and use SSF inside the window to decide who goes first. If we finished the oldest or the second oldest (included the finished request) in the slide window, then we can increment the size of slide window by one (move backward the upper bound by one). If all the requests in the window are finished, then we can go to the next three.

14 -> 13 -> 17 -> 19 -> 16 -> 2 -> 0 -> 15 -> 16

Total distance = 42;

The algorithm I got after we learnt CAP:

We can combine c-scan and cap to satisfies the requirements, which is like c-scan we always go in one direction, once the cap of one station hits three, we will go to that station immediately, and if there are multiple station hit, we go to the one that first submitted the request, and do the following in chronological order.

14 -> 13 -> 6 -> 2 -> 17 -> 16 -> 15 -> 19 -> 0

Total distance = 42;

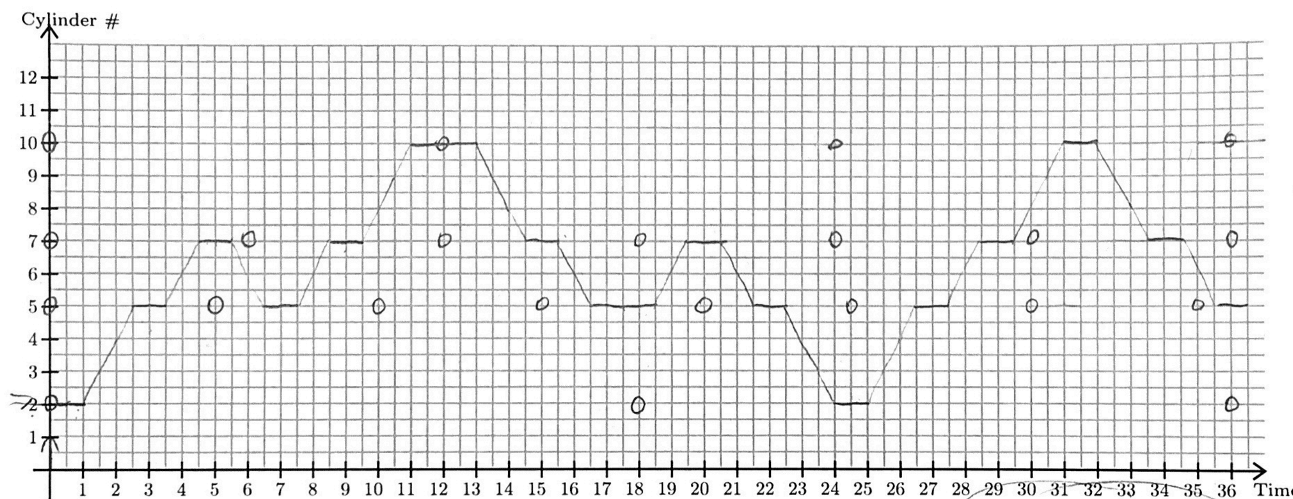
Since the algorithm that combine c-scan and cap can access to the request that are very new and still satisfies the requirement, I think it is better than my first one.

Problem 2

A disk has 12 cylinders and that are concurrently used by 4 processes. Requests coming from process P1 always target cylinder 10 and arrive every 12 time units. Requests form P2 target cylinder 7 and arrive every 6 time units. Requests from P3 target cylinder 5 and arrive every 5 time units. Requests from P4 target cylinder 2 and arrive every 18 time units.
Assume the following: (1) the first request of each task arrives at time 0; (2) the disk head is able to move by TWO cylinders every time unit; (3) when the disk head is in the right position, it takes ONE extra time unit to complete serving a request. During this time, the disk head cannot move; (4) only after a request has been fully served, a new scheduling decision can be made; (5) the disk head is initially positioned at cylinder 2.

non-p

a) Use the grid below to visualize the schedule produced by the Shortest Scan Next scheduling algorithm until time 36. *Carefully* read the 5 rules described above.



18 requests. d=37

b) Use the same style of grid as in the previous Part a) to visualize the schedule produced by the SCAN scheduling algorithm until time 36. Assume that the disk head travel direction at time 0 is UP. *Carefully* read the 5 rules described above. Next page

c) Use the same style of grid as in the previous Part a) to visualize the schedule produced by the C-SCAN scheduling algorithm until time 36. *Carefully* read the 5 rules described above. Next page

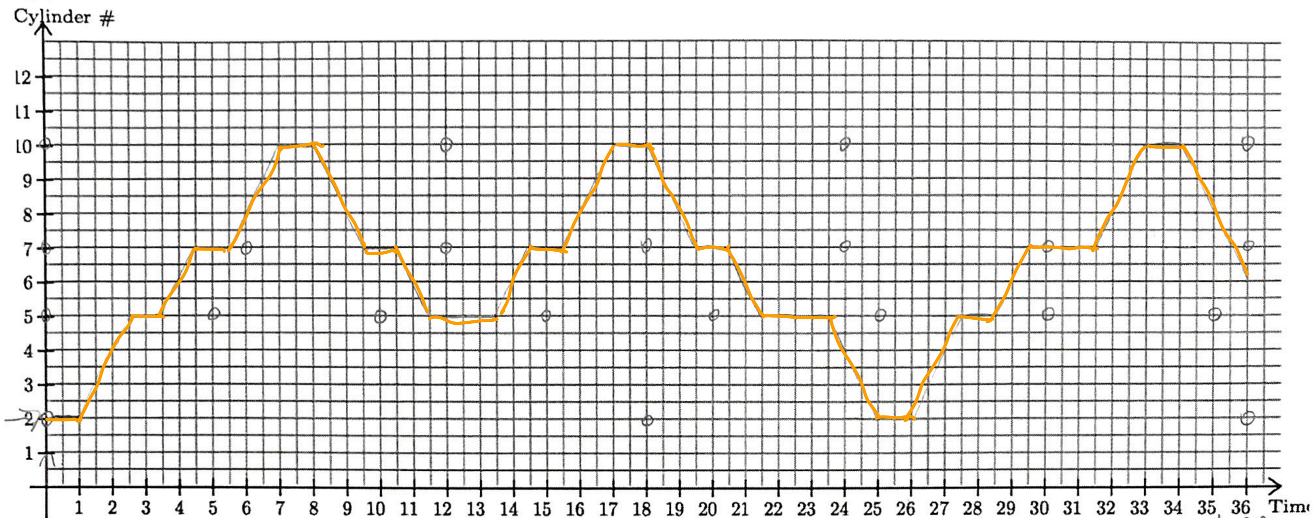
d) Use the same style of grid as in the previous Part a) to visualize the schedule produced by a static priority scheduler that uses the Rate Monotonic rule to assign a fixed priority to each process. Draw only until time 36. *Carefully* read the 5 rules described above. Next page

e) Which algorithm appears to exhibit the best performance if the objective is to minimize the overall distance traveled by the disk head between time 0 and 36 while serving the largest number of requests? Motivate your answer.

SST

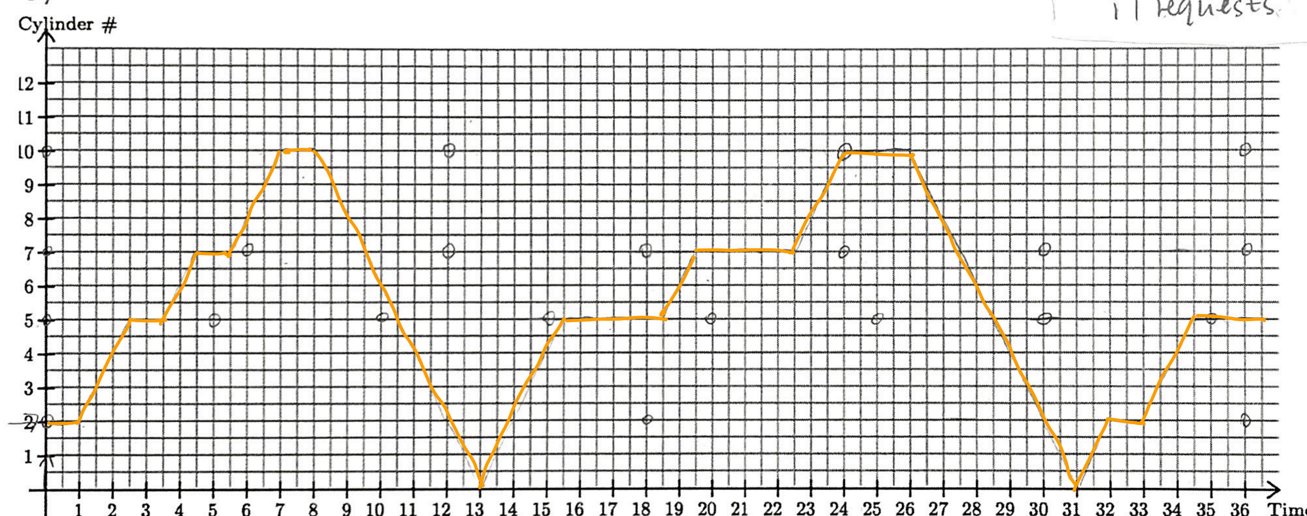
Because number of request: 18 >17 > 16 > 15
and distance: 37 < 38 < 41 < 43
(more reference data see next page)

b).



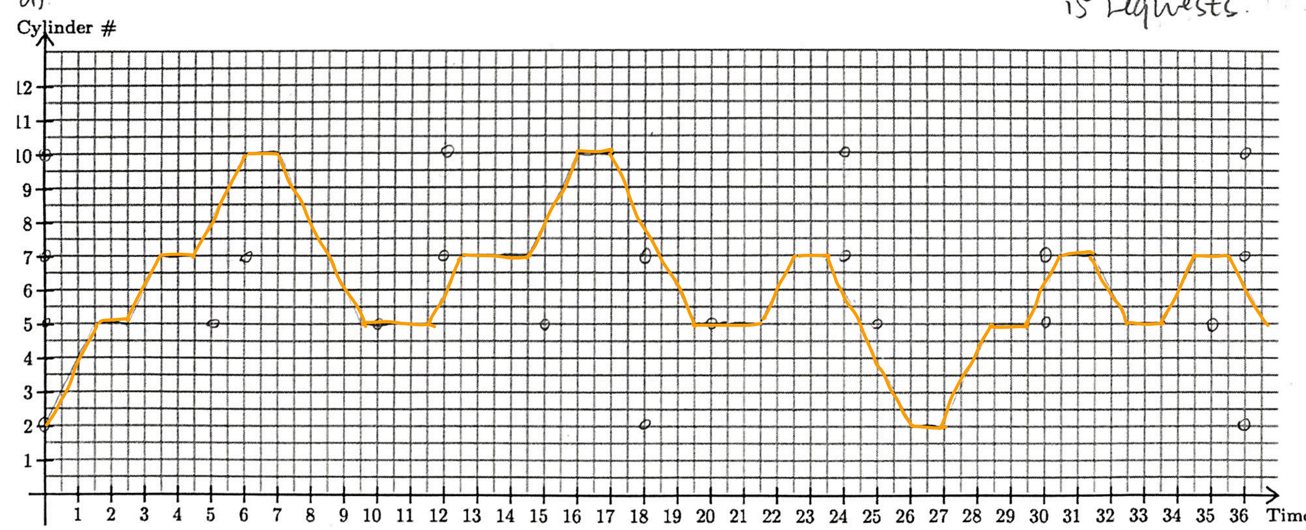
d=38
17 requests.

c).



d=43
15 requests.

d).



16 requests
d=40

Problem 3.

$T_{FI}=7 \quad T_{SR}=9 \quad T_{AP}=12 \quad T_{AD}=17 \quad C_{FI}=1.5 \quad C_{SR}=1.8 \quad C_{AP}=2.3 \quad C_{AD}=3.9$

a). $m=3$

$$\sum_{i=1}^{m} \frac{C_i}{T_i} = \frac{1.5}{7} + \frac{1.8}{9} + \frac{2.3}{12} = 0.606$$

$$m(2^{\frac{1}{m}}-1) = 3(2^{\frac{1}{3}}-1) = 0.780$$

$0.606 < 0.780$

so, schedulable.

b). $m=4$

$$\sum_{i=1}^{m} \frac{C_i}{T_i} = \frac{1.5}{7} + \frac{1.8}{9} + \frac{2.3}{12} + \frac{3.9}{17} = 0.835$$

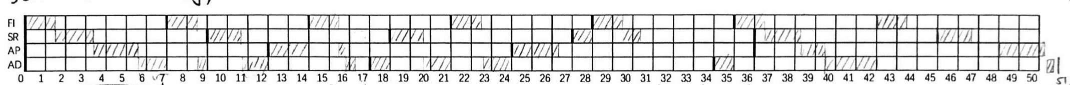
$$m(2^{\frac{1}{m}}-1) = 4(2^{\frac{1}{4}}-1) = 0.757$$

$0.835 > 0.757$

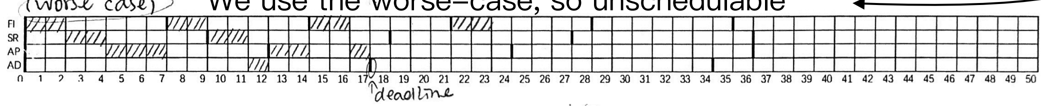
so, no conclusion

b).

RM. schedulable. (avg)

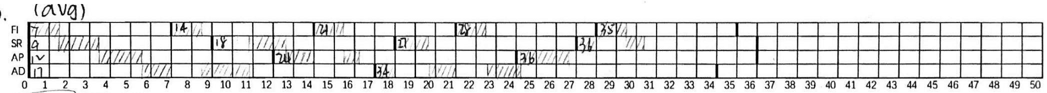


RM. (worse case) We use the worse–case, so unschedulable

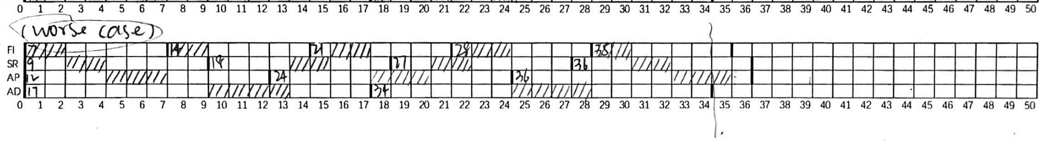


c). $\sum_{i=1}^{m} \frac{C_i}{T_i} = \frac{1.5}{7} + \frac{1.8}{9} + \frac{2.3}{12} + \frac{3.9}{17} = 0.835 < 1$, so schedulable

d). (avg)



EDF (worse case)

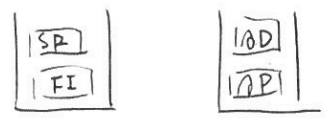


e). Find the LCM of the four tasks’ periods (7, 9, 12, and 17), which is 4284.
Because when the time reach LCM, it is just like “reset” to zero, everything restart again.

f). (next page)

7. f) Consider a 2-CPU processor which is slower than the one considered in the original system. In particular, in the new processor, each task takes 60% more time to execute — but the periods do not change! Is it possible to schedule the system by first splitting the 4 tasks across the 2 CPUs and then using RM on each CPU? Explain your reasoning.

$$\frac{C_i}{T_i} \leq m \cdot (2^{\frac{1}{m}} - 1).$$



1.6. P_1 P_2. 1.6.

$C_{FI} = 1.5(1.6) = 2.4$
$C_{SR} = 1.8(1.6) = 2.88$
$C_{AP} = 2.3(1.6) = 3.68$
$C_{AD} = 3.9(1.6) = 6.24$

① FI $\frac{C_{FI}}{T_{FI}} = \frac{2.4}{7} = 0.343.$

$1(2-1) = 1$

$0.343 < 1$. So put in P_1

② SR $\frac{C_{FI}}{T_{FI}} + \frac{C_{SR}}{T_{SR}} = \frac{2.4}{7} + \frac{2.88}{9} = 0.663$

$2(2^{\frac{1}{2}} - 1) = 0.828$

$0.663 < 0.828$, So put in P_1

③ AP $\frac{C_{FI}}{T_{FI}} + \frac{C_{SR}}{T_{SR}} + \frac{C_{AP}}{T_{AP}} = \frac{2.4}{7} + \frac{2.88}{9} + \frac{3.68}{12} = 0.970$ | $\frac{C_{AP}}{T_{AP}} = \frac{3.68}{12} = 0.307$

4

$3(2^{\frac{1}{3}} - 1) = 0.780$

$0.970 > 0.780$, so don't put in P_1.

$1(2-1) = 1$

$0.307 < 1$, So put in P_2.

④ AD $\frac{C_{FI}}{T_{FI}} + \frac{C_{SR}}{T_{SR}} + \frac{C_{AD}}{T_{AD}} = \frac{2.4}{7} + \frac{2.88}{9} + \frac{6.24}{17} = 1.03 \rightarrow$ No!

$\frac{C_{AP}}{T_{AP}} + \frac{C_{AD}}{T_{AD}} = \frac{3.68}{12} + \frac{6.24}{17} = 0.674$

$2(2^{\frac{1}{2}} - 1) = 0.828$

$0.674 < 0.828$, So put in P_2

So, schedulable.