1

a)

Its not possible for two roommate to starve since there is enough eating utensils for them.

If they eat at the same time, they will need 2 plates, 2 glasses to begin which we have since we start with 4 plates and 3 glasses. Then base on their 2 meal course they choose, they would use up 2 different utensils from the choice of fork, spoon and chopsticks. Since we have 2 of each for fork spoon and chopstick, they can both have the meals without having to wait. Hence, neither of them will starve.

b) They will still not starve. If 4 roommate are in the room, since they wait glass(which is set to 3) only 3 roommate can begin eating the 2 course meal. One of the roommate will always be able to finish the meal, which means that the 4^{th} roommate that is waiting for the glass to start will eventually get his turn once the roommate finish and signal(glass).

c) Since we only have 3 glasses, only 3 roommate can start eating hence and each uses 1 plate, the max amount of dirty plate will also be 3 since that's the highest amount of people that can eat at once.

d) There is no deadlock. There is always one roommate that can finish allowing others to start eating.

e)

assuming roommate 1 –course meal is needs fork and chopstick

assuming roommate 2 –course meal is needs fork and chopstick

assuming roommate 3 –course meal is needs spoons and chopstick

assuming roommate 4 –course meal is needs spoons and chopstick

		Plate	Glasses	Forks	Spoons	Chops stick	
	R(k)	4	3	2	2	2	
p1	C1	1	1	1	0	1	
p2	C2	1	1	1	0	1	
p3	C3	1	1	0	1	1	
p4	C4	1	1	0	1	1	
		Plate	Glasses	Forks	Spoons	Chops stick	
	V(k)	1	0	0	0	0	
p1	A1	1	1	1	0	1	
p2	A2	1	1	1	0	1	
p3	A3	1	1	0	1	0	
p4	A4	0	0	0	0	0	
		Plate	Glasses	Forks	Spoons	Chops stick	
p1	N1	0	0	0	0	0	11101 - 11101 = 00000
p2	N2	0	0	0	0	0	11101 - 11101 = 00000
p3	N3	0	0	0	0	1	11011 - 11010 = 00001
p4	N4	1	1	0	1	1	11011 - 00000 = 11011

We first allocate resources to the first 3 roommate. We can see that roommate 4 cannot take any resources since he's waiting on glass which is 0 at the moment. Roommate 3 has to wait for chopstick to be free since roommate1 and roommate 2 also require the use of chopstick. Hence, roommate 3 and roommate 4 will finish after roommate 1 and 2 finish.

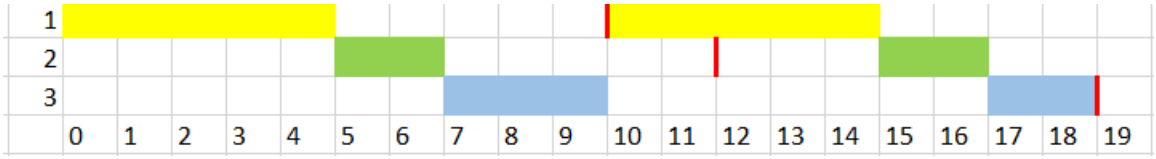
2)

a)

$5/10 + 2/12 + 5/19 <= 3(2^{(1/3)}-1)$

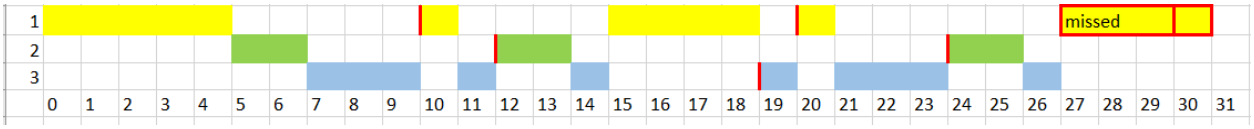
$0.93 <= 0.78$

Cannot conclude if we can schedule under RM based on formula. We will have to plot it to see if its possible.



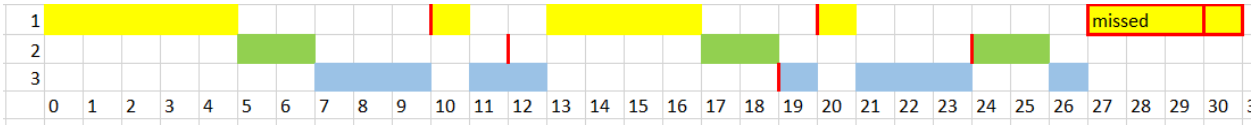
We can see that the task are able to complete under the longest period (19ms), hence we can say that it's RM scheduler.

b)



We will miss task 1 using RM, so its not RM schedulable.

c)



We still miss task 1 using RM, so its not EDF schedulable.

d) If possible we can try to make task 1 and task 3 used different resources so it will be schedulable like we did in 2a. If this is not possible, we can make it so task 2 also share the resources with task 1 and 3 so it cannot interrupt task 1 and 3 and has to wait for the resource so task 1 can complete on time.

3

a)

		R1	R2	R3		R1	R2	R3	
	V(k)	10-7 = 3	5-2=3	7-5=2					
P1	A1	0	1	0	N1	7	4	3	753 - 010 = 743
P2	A2	2	0	0	N2	1	2	2	322 - 200 = 200
P3	A3	3	0	2	N3	6	0	0	902 - 302 = 600
P4	A4	2	1	1	N4	0	1	1	222- 211 = 011
P5	A5	0	0	2	N5	4	3	1	433 - 002 = 431

B)

	work available = 3 3 2	
p1	7 4 3 <= 3 3 2	NOPE
p2	1 2 2 <= 3 3 2	YES
new work	3 3 2 + 2 0 0 = 5 3 2	
	work available = 5 3 2	
p3	6 0 0 <= 5 3 2	NOPE
P4	0 1 1 <= 5 3 2	YES
new work	4 5 4 + 2 1 1= 7 4 3	
	work available = 7 4 3	
p5	431 <= 7 4 3	YES
new work	7 4 3 + 0 0 2 = 7 4 5	
	work available = 7 4 5	
P1	7 4 3 <= 7 4 5	YES
new work	7 4 5 + 0 1 0 = 7 5 5	
p3	6 0 0 <= 7 5 5	YES
new work	7 4 5 + 3 0 2= 10 5 7	

Sequence <P2,P4,P5,P1,P3>

C)

		R1	R2	R3		R1	R2	R3	
	V(k)	10-7 = 3	5-2=3	7-5=2					
P1	A1	0	1	0	N1	7	4	3	753 - 010 = 743
P2	A2	2	0	0	N2	1	2	2	322 - 200 = 200
P3	A3	3	0	2	N3	6	0	0	902 - 302 = 600
P4	A4	2	1	1	N4	0	1	1	222- 211 = 011
P5	A5	0	0	2	N5	4	3	1	433 - 002 = 431
		R1	R2	R3		R1	R2	R3	
	V(k)	10-10 = 0	5-5=0	7-5=2					
P1	A1	0	1	0	N1	7	4	3	753 - 010 = 743
P2	A2	2	0	0	N2	1	2	2	322 - 200 = 200
P3	A3	3	0	2	N3	6	0	0	902 - 302 = 600
P4	A4	2	1	1	N4	0	1	1	222- 211 = 011
P5	A5	3	3	2	N5	1	0	1	433 - 332 = 101

Since V(k) is 0 0 2 , it's not enough to complete any of the process to free up more resources so it enters a deadlock. This allocation is should not be granted.