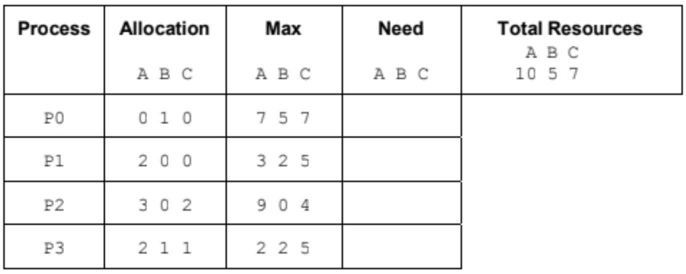
1) Consider a system with twelve resources and three threads: T_0, T_1, and T_2. Thread T_0 requires ten resources, thread T_1 may need as many as four, and thread T_2 may need up to nine resources. Suppose that, at time t_0, thread T_0 is holding five resources, thread T_1 is holding two resources, and thread T_2 is holding two resources

   (a) Is system a t0 in safe state?
   (b) Suppose that, at time t_1 , thread T_2 requests and is allocated one more resource. Is system in safe state now?

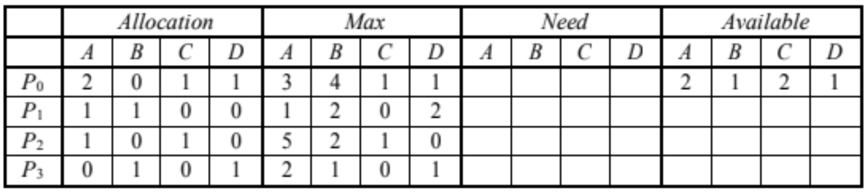
2) Suppose that a system is in an unsafe state. Show that it is possible for the threads to complete their execution without entering a deadlocked state.

3) Show that the system is in a safe state by demonstrating an order in which the processes may complete.



Process	Allocation A B C	Max A B C	Need A B C	Total Resources A B C 10 5 7
P0	0 1 0	7 5 7		
P1	2 0 0	3 2 5		
P2	3 0 2	9 0 4		
P3	2 1 1	2 2 5		

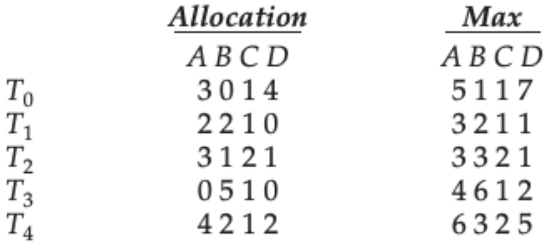
4) Consider the following snapshot of a system about four processes and four resources A, B, C and D. At the current moment assume that (after the allocations are made) the available resources are 2 instances of resource A, 1 instance of resource B, 2 of resource C, 1 of resource D. Is the system in a safe state or is it deadlocked? Why or why not?



	Allocation				*Max*				*Need*				*Available*			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*
P_0	2	0	1	1	3	4	1	1					2	1	2	1
P_1	1	1	0	0	1	2	0	2								
P_2	1	0	1	0	5	2	1	0								
P_3	0	1	0	1	2	1	0	1								

5) Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe

   a) Available = (0, 3, 0, 1)
   b) Available = (1, 0, 0, 2)



	Allocation *A B C D*	*Max* *A B C D*
T_0	3 0 1 4	5 1 1 7
T_1	2 2 1 0	3 2 1 1
T_2	3 1 2 1	3 3 2 1
T_3	0 5 1 0	4 6 1 2
T_4	4 2 1 2	6 3 2 5