由题可知，T_0时刻的资源分配情况：

资源情况 / 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P_1	5	5	9	2	1	2	3	4	7			
P_2	5	3	6	4	0	2	1	3	4			
P_3	4	0	11	4	0	5	0	0	6	2	3	3
P_4	4	2	5	2	0	4	2	2	1			
P_5	4	2	4	3	1	4	1	1	0			

(1) 利用安全性算法对T_0时刻的资源分配情况进行分析：

资源情况 / 进程	Max			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P_3	2	3	3	0	0	6	4	0	5	6	3	8	true
P_4	6	3	8	2	2	1	2	0	4	8	3	12	true
P_5	8	3	12	1	1	0	3	1	4	11	4	16	true
P_1	11	4	16	3	4	7	2	1	2	13	5	18	true
P_2	13	5	18	1	3	4	4	0	2	17	5	20	true

可知，在T_0时刻存在一个安全序列{P_3,P_4,P_5,P_1,P_2}，所以系统是安全的。

(2) P_2发出请求向量$Request_2(0,3,4)$，系统按银行家算法进行检查：

$Request_2(0,3,4) \leqslant Need_2(1,3,4)$

$Request_2(0,3,4) > Available_2(2,3,3)$

不予分配，让P_2等待。

(3) P_4发出请求向量$Request_4(2,0,1)$，系统按银行家算法进行检查：

$Request_4(2,0,1) \leqslant Need_4(2,2,1)$

$Request_4(2,0,1) \leqslant Available_4(2,3,3)$

系统先假定可为P_4分配资源，并修改Available，$Allocation_4$和$Need_4$向量，由此形成的资源变化情况如图所示：

资源情况 / 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P_1	5	5	9	2	1	2	3	4	7			
P_2	5	3	6	4	0	2	1	3	4			
P_3	4	0	11	4	0	5	0	0	6	0	3	2
P_4	4	2	5	4	0	5	0	2	0			
P_5	4	2	4	3	1	4	1	1	0			

再利用安全性算法检查此时系统是否安全：

资源情况 / 进程	Max			Need			Allocation			Work+Allocation			Finish
	A	B	C	A	B	C	A	B	C	A	B	C	
P_3	0	3	2	0	0	6	4	0	5	4	3	7	true
P_4	4	3	7	0	2	0	4	0	5	8	3	12	true
P_5	8	3	12	1	1	0	3	1	4	11	4	16	true
P_1	11	4	16	3	4	7	2	1	2	13	5	18	true
P_2	13	5	18	1	3	4	4	0	2	17	5	20	true

可以找到一个安全序列{P_3,P_4,P_5,P_1,P_2}。因此系统是安全的，可以立即将 P_4 所申请的资源分配给它。

(4) P_1 发出请求向量 $Request_1(0,2,0)$，系统按银行家算法进行检查：

$Request_1(0,2,0) \leq Need_1(3,4,7)$

$Request_1(0,2,0) \leq Available_1(0,3,2)$

系统先假定可为 P_1 分配资源，并修改 Available，$Allocation_1$ 和 $Need_1$ 向量，由此形成的资源变化情况如图所示：

资源情况 / 进程	Max			Allocation			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P_1	5	5	9	2	3	2	3	2	7			
P_2	5	3	6	4	0	2	1	3	4			
P_3	4	0	11	4	0	5	0	0	6	0	1	2
P_4	4	2	5	4	0	5	0	2	0			
P_5	4	2	4	3	1	4	1	1	0			

可知，可用资源 Available(0,1,2)已不能满足任何进程的需要，所以系统进入不安全状态，此时系统不分配资源。