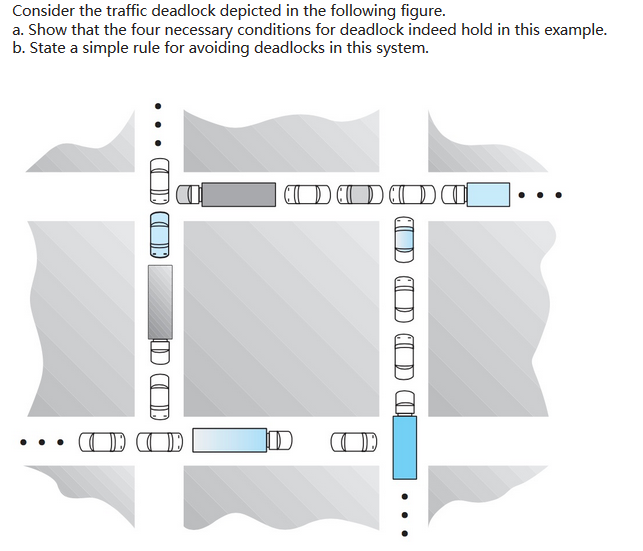
1. 

**ANS:**

(1) 4个必要条件： 互斥；持有并等待；无抢占；循环等待

互斥：道路的每个路段（如十字路口的中心）一次只允许一辆车通过

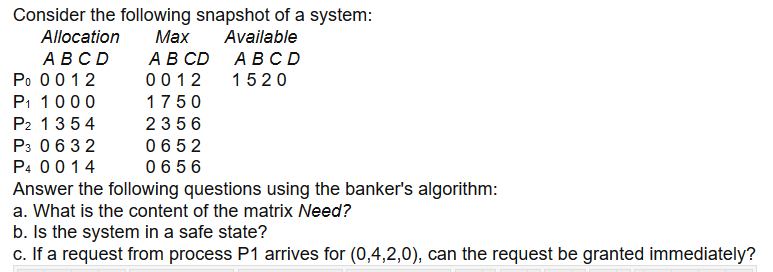
持有并等待：位于道路每个路段的车辆在等待前进方向车辆所在的路段（即车辆需要前进，但前进方向上已经有车辆）

无抢占：位于道路上每个路段的车辆不能抢占前进方向上车辆的位置，只能等待前面的车辆移动后释放该路段

循环等待： 由于图中十字路口的存在，导致图中离十字路口最近，但还未占有十字路口的车辆在等待与其前进方向成90度的正在占用十字路口的车辆，从而造成了一次等待方向的转变，在经过4个这样的十字路口之后，产生了一个循环等待的情况

图中的以上四个情况满足死锁的4个必要条件，在这四个情况同时发生时，死锁产生。

（2） 一个简单的方法是：不允许有车辆长期占用着十字路口中心，且规定某个方向（如相对右边）的车辆总是先行

**2.** 

ANS：

（1）Need矩阵

     A   B   C   D

P0     0    0    0   0

P1     0    7    5   0

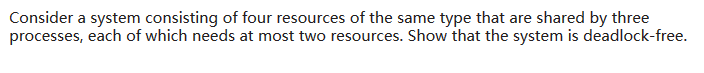
P2     1    0    0   2

P3     0    0    2    0

P4    0    6    4   2

（2）是安全状态，序列 {P0,P2,P3,P4,P1}满足执行条件<p0

(3) 可以被立刻分配，因为Request(P1)< Need(P1) , Request(P1) < Available,且修改状态后，新的资源分配状态是安全的

3. 

ANS：

假设该系统产生了死锁，则因为共有4个资源及3个进程，说明其中一个进程必定同时占有了2个资源。死锁产生的4个必要条件之一是占有并等待，即一个进程需要占有资源并等待另一个被其他进程占有的资源，此系统中已经占有2个资源的进程不需要额外资源，因此该系统也不满足循环等待的条件。综上，该系统未能同时满足死锁产生的4个必要条件，因此该系统是deadlock-free的