P260-P261:

4,9,10,13,14,15,

20,21,22，24,26

4、资源死锁的四个条件在每个资源只有一个实例时才是充分的。考虑哲学家就餐问题，如果某个叉子的位置上多放一把叉子，就不会发生死锁。

9、意味着请求永远无法被满足。

10、图中的模型能够表示出在哪个区域会进入死锁，而安全状态能够保证不会进入死锁，不安全状态则无法保证。

13、有可能出于既不安全又不死锁的状态。

考虑以下情况：

|  |  |  |
| --- | --- | --- |
|  | 已有数量 | 最大需求 |
| A | 4 | 9 |
| B | 2 | 4 |
| C | 2 | 7 |

空闲：2

此时如果没有任何一个进程释放资源，则无法完成所有的进程。

但是如果进程A释放一个资源，则死锁就不会发生。

14、根据银行家算法拒绝这个请求并不意味着接受这个请求就会进入死锁，而是会进入不安全状态，进入不安全状态不一定会导致死锁。

15、无法设计这样的算法。因为无法根据进程最大资源需求信息来判断是否死锁可能发生。

20、当 n 小于6时，都存在一个进程能够先占有两台磁带机，释放资源后就能使得其他进程能够顺利完成，死锁不会发生。n=6时，如果每个进程分配得到一个磁带机，就会导致死锁。

21、检测一遍有没有被满足的资源数均小于或等于A，需要进行 mn 次操作；总共进行 n次检验，所以总共是 次操作。即 a=1, b=2。

22、需求矩阵：

A 0 1 0 0 2

B 0 2 1 0 0

C 1 0 3 0 0

D 0 0 1 1 1

x=1时，D 进程能被满足，释放资源后 A=1 1 2 2 1

但是还是导致死锁的产生，只有 C 进程是只因为第三个资源不足而无法完成，

如果 x=2，完成 D 进程之后，A=1 1 3 2 1，

C 进程能被完成，之后 A=2 2 3 3 1，

B 进程能够完成，之后 A=4 2 4 4 1，

但是进程 A 无法完成。因为资源5的最大需求为3，高过了已分配资源与可用资源中的资源5的总和。所以无论 x 为多少，都无法保证安全状态。

24、假设进程 A 的请求顺序是 123；

只要 B首先申请的资源是1，那么 A 或 B 其中之一会先开始运行直到释放所有资源，避免了死锁发生。反之，如果 B 先申请得到了2或3，A同时获得了1，那么两个进程都无法完成。

进程 B 的请求顺序以及死锁发生情况：

|  |  |
| --- | --- |
| 123 | 不会 |
| 132 | 不会 |
| 213 | 会 |
| 231 | 会 |
| 312 | 会 |
| 321 | 会 |

可以看出不发生死锁的概率是1/3。

26、可以对所有账户进行编号，进行一次转账时先锁定编号较小的账户。这样只要一次转账中锁定了一个账户后另外一个账户编号足够大就能保证该进程顺利完成，从而释放资源，避免了循环等待。