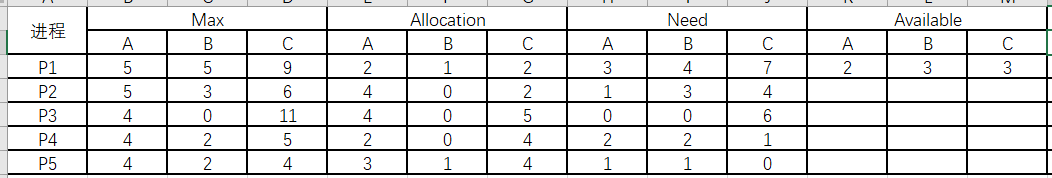
银行家问题:

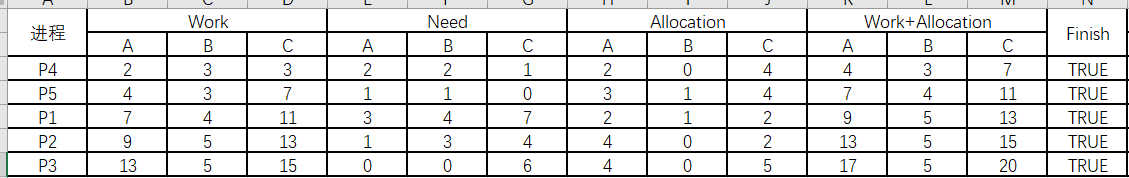


1. 在T0时刻，资源分配如下表（Need=Max-Allocation）:



初始work=2 3 3

利用安全性算法对资源分配情况进行分析如下表



在T0时刻存在一个安全序列{P4,P5,P1,P2,P3}

1. P2：Request(0, 3, 4)，根据银行家算法：

Request(0, 3, 4) ≤ Need(1, 3, 4) （满足）

Request(0, 3, 4) > Available(2, 3, 3) （不满足）

所以不能分配，需等待。

1. P4：Request(2, 0, 1)，根据银行家算法：

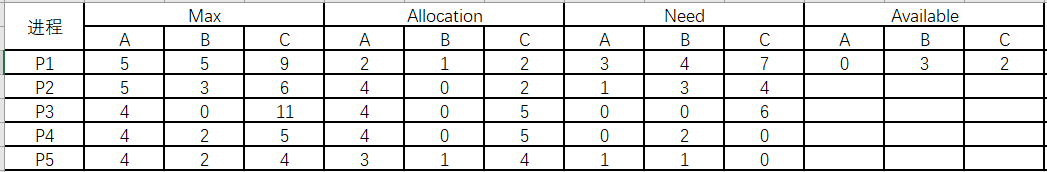
Request(2, 0, 1) ≤ Need(2, 2, 1) （满足）

Request(2, 0, 1) ≤ Available(2, 3, 3) （满足）

假设接受资源请求：

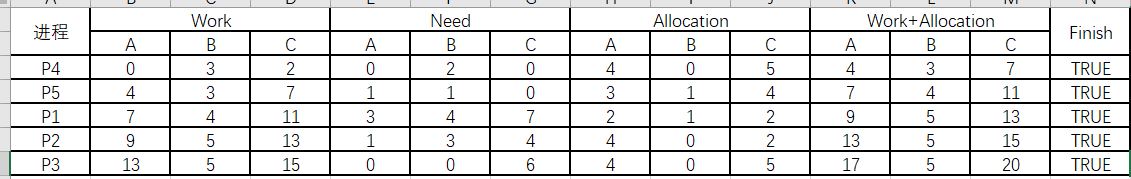
（Allocation += Request）（Need -= Request）（Available-=Request）

（P2不变化）



初始work=0 3 2

利用安全性算法对资源分配情况进行分析如下表



存在一个安全序列{P4,P5,P1,P2,P3}，所以可以向P4分配资源。

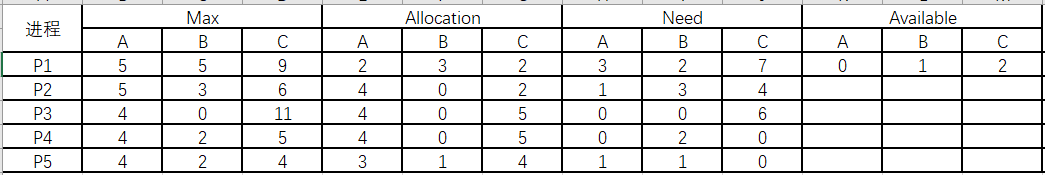
1. P1：Request(0, 2, 0)，根据银行家算法：

Request(0, 2, 0) ≤ Need(3, 4, 7) （满足）

Request(0, 2, 0) ≤ Available(0, 3, 2) （满足）

假设接受资源请求：

（Allocation += Request）（Need -= Request）（Available-=Request）



初始work=0 1 2，任何一个进程都不满足need ≤ work，所以不处于安全状态，不能分配。