答：(1)

| 最大需求 | 已分配 | Need |

P1 | 5 5 9 | 2 1 2 | 3 4 7 |

P2 | 5 3 6 | 4 0 2 | 1 3 4 |

P3 | 4 0 11 | 4 0 5 | 0 0 6 |

P4 | 4 2 5 | 2 0 4 | 2 2 1 |

P5 | 4 2 4 | 3 1 4 | 1 1 0 |

且根据已分配和A(17),B(5),C(20)可以求出work = (2 3 3)

P4 Need < work 给P4分配,work = work + 已分配 = (4 3 7)

P2 Need < work 给P2分配,work = work + 已分配 = (8 3 9)

P3 Need < work 给P3分配,work = work + 已分配 = (12 3 14)

P5 Need < work 给P5分配,work = work + 已分配 = (15 4 18)

P1 Need < work 给P1分配,work = work + 已分配 = (17 5 20)

故: |最大需求|已分配 | Need | w+a |finish|

P4 | 4 2 5 | 2 0 4 | 2 2 1 | 4 3 7 | T |

P2 | 5 3 6 | 4 0 2 | 1 3 4 | 8 3 9 | T |

P3 | 4 0 11| 4 0 5 | 0 0 6 | 12 3 14 | T |

P5 | 4 2 4 | 3 1 4 | 1 1 0 | 15 4 18 | T |

P1 | 5 5 9 | 2 1 2 | 3 4 7 | 17 5 20 | T |

可得T0时刻为安全状态,安全系列为: P4， P2， P3 ，P5， P1

(2)

P2:Request(0,3,4) > work(2 3 3)

故P2:Request(0,3,4)时不能分配

(3)

P4:Request(2,0,1) < work(2 3 3)

更新资源分配表:

| 最大需求 | 已分配 | Need |

P1 | 5 5 9 | 2 1 2 | 3 4 7 |

P2 | 5 3 6 | 4 0 2 | 1 3 4 |

P3 | 4 0 11 | 4 0 5 | 0 0 6 |

P4 | 4 2 5 | 4 0 5 | 0 2 0 |

P5 | 4 2 4 | 3 1 4 | 1 1 0 |

此时 work = (0 3 2)

P4 Need < work 给P4分配,work = work + 已分配 = (4 3 7)

P2 Need < work 给P2分配,work = work + 已分配 = (8 3 9)

P3 Need < work 给P3分配,work = work + 已分配 = (12 3 14)

P5 Need < work 给P5分配,work = work + 已分配 = (15 4 18)

P1 Need < work 给P1分配,work = work + 已分配 = (17 5 20)

故:| 最大需求 |已分配 | Need | w+a |finish|

P4 | 4 2 5 | 4 0 5 | 0 2 0 | 4 3 7 | T |

P2 | 5 3 6 | 4 0 2 | 1 3 4 | 8 3 9 | T |

P3 | 4 0 11 | 4 0 5 | 0 0 6 | 12 3 14 | T |

P5 | 4 2 4 | 3 1 4| 1 1 0 | 15 4 18 | T |

P1 | 5 5 9 | 2 1 2| 3 4 7 | 17 5 20 | T |

可得T0时刻为安全状态,安全系列为: P4， P2， P3， P5， P1

故P4:Request(2,0,1)时可以分配

(4)

在(3)中，work已更新为(0 3 2)

P1:Request(0,2,0) < work(0 3 2)

更新资源分配表:

| 最大需求 | 已分配 | Need |

P1 | 5 5 9 | 2 3 2 | 3 2 7 |

P2 | 5 3 6 | 4 0 2 | 1 3 4 |

P3 | 4 0 11 | 4 0 5 | 0 0 6 |

P4 | 4 2 5 | 4 0 5 | 0 2 0 |

P5 | 4 2 4 | 3 1 4 | 1 1 0 |

此时 work = (0 1 2)

P1-P5所有的 Need 都大于 work,已不能满足任何进程的需要,

系统进入不安全状态，因此不能分配资源。