1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：prefixspan算法的缺陷是什么？

讨论后的理解：PrefixSpan运行时最大的消耗在递归的构造投影数据库。如果序列数据集较大，项数种类较多时，算法运行速度会有明显下降

1. 提出的问题2：使用古典关联规则进行推荐的好处是什么？

讨论后的理解：规则的右侧可以是任何商品，而传统的分类算法中，只能针对一组预先定义好的类别，因此不太适合做推荐系统。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. 问题3：P41最后一行，经过算法计算之后得到的序列模式集合为包括{<{30}，{40}>,<{40}，{30}>}这样的序列，算不算是矛盾呢？需不需要考虑删除呢？

自己的理解：：一个项在项集中最多出现一次，但是一个时间可以在序列中出现多次。 而且次序在序列中很重要，但是在项集中不重要。如{1, 2}和{2, 1}表示同一个项集，但是<i1i2><i1i2>和<i2i1><i2i1>表示不同序列。所以{<{30}，{40}>,<{40}，{30}>}是不算矛盾的，不需要删除。

1. 问题4：r—PrefixSpan()与PrefixSpan最大不同点是什么？

自己的理解：在每次递归调用中，或者前缀本身必须包含ik或者映射数据库中的每个序列必须包含ik，因为这个函数仅找出那些包含ik的频繁序列。

1. （必填）读书计划

1、本周完成的内容章节：如2.8-2.9，3.5

2、下周计划：3.6-3.7、

读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

本次阅读内容主要在于序列模式挖掘，主要用到prefixspan算法：先找出单个的频繁项，再找出以它为后缀的频繁模式，进而通过递归，最终生成频繁序列。2.9是本章最后一节，总结了之前的算法，做到从序列模式中产生规则。