读书报告内容：（复制下面部分到一个新文档，名称命名为自己的名字，填写完交给组长，没做第四部分可以不复制第四部分）

1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：P40X写到Prefixspan算法的主要优点是不生成候选序列，但在每一次对映射数据库进行匹配时，都需要遍历一次序列集，从时间复杂度上考虑，是不是并没有优于GSP呢？

讨论后的理解：其实深度搜索相比于广度搜索，可能并没有在时间上简洁多少，但是它不需要同时储存信息，这样在空间复杂度上，Prefixspan算法是更优的。

1. 提出的问题2： 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？

讨论后的理解：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. （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：r-prefixspan()与Prefixpan最大的不同点是什么？

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。

1. 问题4：Prefixspan算法的优缺点

自己的理解：Apriori 算法的优点

1.对原始数据要求很低；

2. 这种算法思路比较清晰直接，实施起来比较简单。

Apriori 算法的缺点

1.Aprior 算法只适合发现短的频繁模式，对长频繁模式比较困难；

2.Aprior 算法需要不断扫描数据库从中寻找候选集，涉及大量I/O 操作；

3.每一步产生侯选集时循环产生的组合过多，未排除不应该参与组合的元素。

1. （必填）读书计划

1、本周完成的内容章节：如2.8-2.9

2、下周计划：3.6-3.7

四、（选做）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）

2.8基于 prefixspan算法的序列模式挖掘

prefixspan算法流程：

1）找出所有长度为1的前缀和对应的投影数据库

2）对长度为1的前缀进行计数，将支持度低于阈值α的前缀对应的项从数据集S删除，同时得到所有的频繁1项序列，i=1.

3）对于每个长度为i满足支持度要求的前缀进行递归挖掘

PrefixSpan算法由于不用产生候选序列，且投影数据库缩小的很快，内存消耗比较稳定，作频繁序列模式挖掘的时候效果很高。比起其他的序列挖掘算法比如GSP,FreeSpan有较大优势，因此是在生产环境常用的算法。

2.9从序列模式中生成规则

序列规则（Sequential Rules）：可表述为蕴含式形式：X→ Y，X是Y的一个子序列，支持度和置信度类似先前定义。给定minsuo和minconf，可以从序列模式中得到序列规则。

标签序列规则（Label Sequential Rules）：可表述为蕴含式形式：X → Y ，X是将Y中部分项目替换成通配符得到的序列。替换掉的项目称作标签（Label）。

分类序列规则（Class Sequential Rules）：类似于CAR，可表述为蕴含式形式：X→ y，y为Y元素，X为序列，Y为分类标记集合。