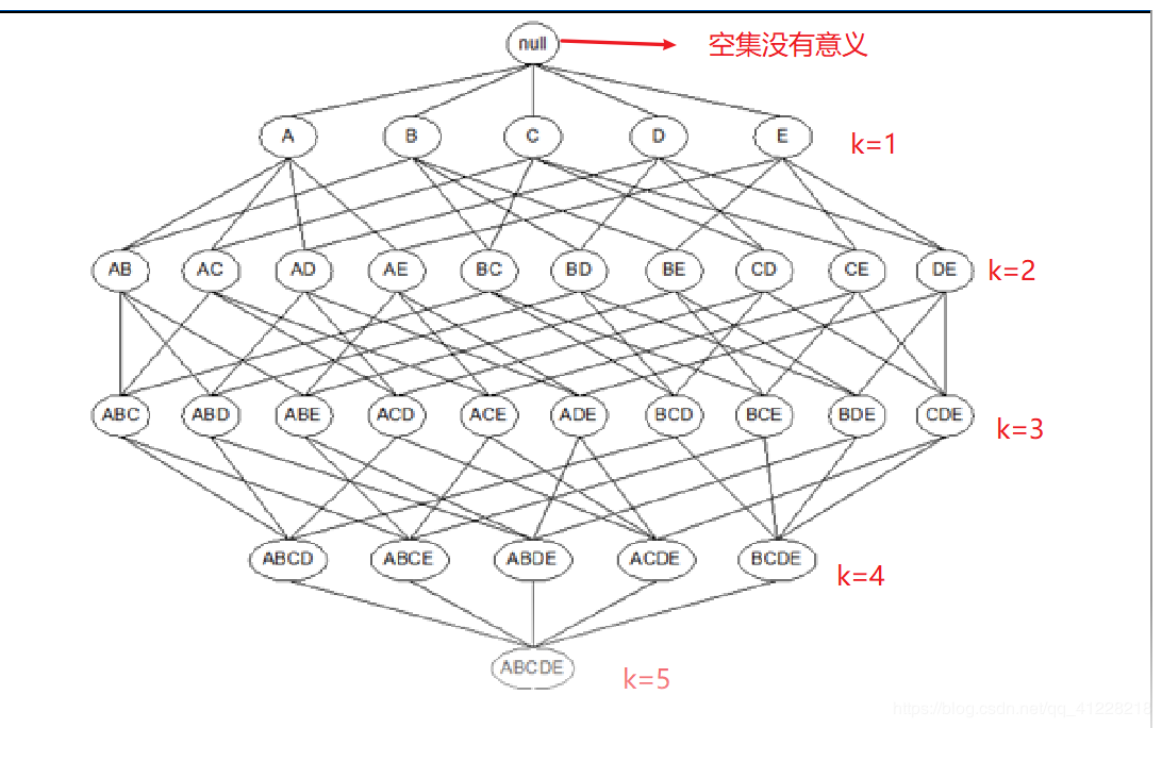
1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：P23为什么“不把同时含有频繁项目和稀有项目的项集作为频繁项目生成”？

讨论后的理解：这样既能避免生成众多无意义的关联规则，又能顺利地对稀有项目进行挖掘。如书中的例子{面包、鸡蛋、牛奶、平底锅}组成的频繁项目集是没有意义的，因为在每一个食品店，面包、牛奶、鸡蛋这类商品在每位客户的每次采购中都会购买。

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2：P18怎么得到Apriori的算法是指数级的？

自己的理解：Apriori的算法的时间复杂度是2的n次方，这是二项式定理的逆运算，如果项目的数目是n，那么apriori算法生成的项集数量是，下图为项目数量为5的例子，生成项集32个



1. 问题3：P19最上方置信度为什么采用f.count/(f-a).count

自己的理解：

1. 问题4：p22第二段，提到为了解决稀有项目问题，一个常见的解决办法是将数据分为几个小块，每个子集只包含哪些出现频率比较接近的项目。关于这个分块方法不是很理解

自己的理解：这个是将不同频率区间的项目分到一个子集里面，比如0-0.2，0.2-0.4，0.4-0.6，0.6-0.8，0.8-1.0，根据这个频率区间分为5个子集，然后每个子集给定一个最小项目支持度，来解决无意义关联规则和组合爆炸的问题。

1. 问题5：感觉apriori太暴力了，有没有一些其他的优于apriori的算法

自己的理解：其实书本P18提到了一个FP-growth算法，是韩家炜老师提出的挖掘频繁项集的方法，是将数据集存储在一个特定的称作FP树的结构之后发现频繁项集或频繁项对，即常在一块出现的元素项的集合FP树。

FP-growth算法比Apriori算法效率更高，在整个算法执行过程中，只需遍历数据集2次，就能够完成频繁模式发现。

三、（必填）读书计划

1、本周完成的内容章节：如2.1-2.4

2、下周计划：第二章看完（2.5-2.9）

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）

2.1关联规则的基本概念

关联规则挖掘的目的是在数据项目中找出所有的并发关系。目标是找出事务几何T中所有满足支持度和置信度分别高于最小支持度（minsup）和最小置信度（minconf）的关联规则。其中支持度是指T中包含的事务的百分比，规则的支持度计算如下：

支持度=

置信度是指既包含了X又包含了Y的事务的数量的数量占所有包含了X的事务的百分比，置信度计算如下：

置信度=

2.2Apriori算法

Apriori算法是种挖掘关联规则的频繁项集算法，一种最有影响的挖掘布尔关联规则频繁项集的算法。该算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用第1步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了[递归](https://baike.baidu.com/item/%E9%80%92%E5%BD%92)的方法。

2.3关联规则挖掘的数据格式

挖掘关联规则的主要的数据类型是事务数据，纯文本文件也可以看作是事务数据，每个文档是一个事务，单词看作是项目。关联规则也可以在关系数据表上进行，只需要将表格数据转换成事务数据，

2.4多最小支持度的关联规则挖掘

Ms-Apriori算法采用另外一种办法，既然统一的支持度值不能兼顾所有的情况，可以设置多个支持度值，每个种类项都有一个最小支持度阈值，然后一个频繁项的最小支持度阈值取其中项集元素中的最小支持度值作为该项集的最小支持度值。这样的话，如果一个频繁项中出现了稀有项集，则这个项集的最小支持度值就会被拉低，如果又有某个项都是出现频率很高的项构成的话，则支持度阈值又会被拉高。当然，如果出现了一个比较难变态的情况就是，频繁项中同时出现了稀有项和普通项，我们可以通过设置SDC支持度差别限制来限制这种情况的发生，使之挖掘的频繁项更加的合理。