测试时间为2小时，请使用任意语言完成，注意增加适当的注释。在六道题目中任选两道即可（注意：前三道题较难，后三道题较简单，自由选择你能做的即可），可邮件或电话联系我：brandenwang@tencent.com，15071086632。

1、实现倒排索引。

原始数据例如：

上表是简化的物料包装数据。字段解释：订单号确定一个包装订单，箱代码表示物料由哪种箱子包装，物料编码表示一种物料，包装数量表示这种物料在对应箱子里的数量。

|  |  |  |  |
| --- | --- | --- | --- |
| 订单号packno | 箱代码boxcode | 物料编码itemcode | 包装数量itemqty |
| 27477860001A | R16 | 02311QYD | 5 |
| 27477860001A | R16 | 0235G6AS | 5 |
| 27477860001A | R16 | 4043491 | 5 |
| 27477860001A | R18 | 02311QYD | 10 |
| 27477860001A | R18 | 0235G6AS | 10 |
| 27477860001A | R18 | 4043491 | 10 |
| 620006310005A | BM09 | 21070343 | 1 |
| 620006310005A | BM09 | 4043491 | 20 |
| 620006310005A | BM09 | 21140701 |  |
| 620006310005A | BM09 | 21170397 | 0.1 |
| 620006310005A | BM09 | 21240373 | 0.5 |

需要实现的倒排索引格式为：

* key：(itemcode，itemqty)，这里物料的数量，指的是每种物料在每个订单里包含的准确数量，而非每个箱子
* value：[packno]

即找出每种物料在哪些订单中出现过，比如物料item1在订单001A中包含了10件，在订单002A中也包含了10件，则倒排索引为(‘item1’, 10.0), [‘001A’, ‘002A’]。

注意：程序需要能处理原始数据中的异常。

**要求：**

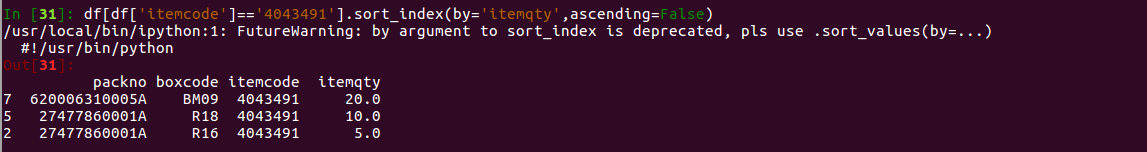
**函数名：gen\_inverted\_index()**

**输入：原始数据文件a.txt**



**输出：倒排索引文件b.txt**

****

****

2、求马里奥能吃到的最大金币数

我们假设一个场景：存在一块二维地图（由一个二维数据表示），地图的每个位置放置了一枚金币，数组的每个数字代表点的高度，马里奥可以上下左右四个方向前进，每次跳一格，且可以从任何初始位置开始，但是只能往高处跳跃吃金币——即只能从1->2，2->3，而不能1->1或2->1，求马里奥能吃到的最大金币数。

下面是一张示例地图：

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

**要求：**

**函数名：mario\_coin\_num()**

**输入：随机生成的地图数据map\_data，List[List[Int]]**

**输出：最大金币数max\_coin\_num，马里奥走的路径mario\_path**

3、缺陷分类

在实际的硬件制造装配过程中，操作员如果发现产品质量有缺陷，会写下缺陷的**现象描述**，如“在装配（前）时发现拉手条来料少了导光柱、定位柱，螺钉孔不良，扳手松动等问题”，后续工程师会对此缺陷进行**分类**，如上述**缺陷类别**为“物料问题”。

但实际过程中工程师只能对部分现象描述进行人工分类，因此需要通过机器学习等方法进行**自动化分类**。

示例数据如下：



包含三类缺陷：物料问题（100条）、结构件外观缺陷（50条）和设计问题（20条）——这里已经对实际问题进行了简化，实际标注数据很不均衡。

**要求：**

**1） 编写程序对原始数据进行数据测量，观测类别之间的区分度——具体使用的测量指标请自定（比如cos距离等）。**

**2） 编写程序对原始数据进行自动化分类，可以使用机器学习方法，也可以使用其他。**

4.在pattern mining的序列模式挖掘算法中常常会涉及子序列的判断，我们对此可抽象成如下判断算法：

例如对于序列A和B，各元素之间存在严格的时序关系，分别用两个list来表示，：

A： strA-0, strA-1,strA-2，strA-3..,strA-n

B: strB-0, strB-1,strB-2，strB-3...,strB-m

( 其中有m >=n)

如果序列A是序列B的子序列，当且仅当:

strA-0是strB-j0的子字符串，strA-1是strB-j1的子字符串，strA-2是strB-j3的子字符串..... strA-i是strB-ji的子字符串...strA-n是strB-jn的子字符串。

其中0<=j0<j1<j2.....<jn<=m.

**请用尽量用高效优雅的python代码实现子序列判断算法，下面是方法头：**

**def is\_subsequence(list\_a, list\_b):**

**#判断list\_A是否是list\_B的子序列，如果是返回true,否则返回false**

**return true**

1. 很多应用都会使用配置文件，配置项一般是“=”号分隔key和value，如 name =zhangsan，一行最多一个配置项。一般使用“#”作为注释的标识符，注释可以是整行，也可能出现在行尾。**请编写一个简单的配置文件解释器，输入为配置文件名，输出为在屏幕上打印出key、value值。**

6、编写一个程序，给定一个本地目录，在该目录（可能包含嵌套子目录）下所有文件内容中匹配形如'tencent-整数'的单词，如tencent-01','tencent-100'等，**将所有这些单词后缀的整数相加得出一个累加值，并返回这个累加值。**

注： 给定目录下的文件数量可能很多，文件大小可能很大，可考虑使用并发，既可以使用python的缺省包和第三⽅包，也可以使用第三方框架。