移动群组消费现阶段任务：

　　现实生活中，消费常常以小团体为单位的形式出现，我们的任务是识别某个消费场所（如某个大型商场）的消费人群中以团体为消费单位的人群，也就是找出哪些人是在同一群组的。

　数据来源于模拟器，模拟器模拟许多人在某个商场的行为，这些人在商场里面“行走”的同时会被商家吸引，从而坐标位置会发生变化，同一群组的人他们的坐标在大部分时刻会比较“紧密”。

　根据坐标的这一特点，我们可以用聚类算法对人的坐标进行聚类，从而将关系密切的人聚到同一类，达到识别目的。模拟器在产生数据时实际已经设定好每个人所属的类，即哪些人属于同一个类。我们用算法聚类的结果将对比真实的设定，计算聚类算法的准确度。

数据格式：

　　模拟器会产生TXT文件，这些TXT文件包含了所有人的坐标信息。一行表示的是在某个时刻所有人的X,Y坐标，不同行代表不同时刻的坐标信息。

数据格式大致如下：人的预设组号 此人的x坐标 此人的y坐标.....

1 37.2 23.4 2 34.5 36.3 3 12.89 35.0 4 ........

1 39.3 25.0 2 33.1 37.8 3 15.4 34.7 4 ........

. . .

. . .

. . .

进展：

1. 实现了ｋｍｅａｎｓ算法，改进后的Ｋｍｅａｎｓ算法，完成了Ｋｍｅａｎｓ算法的演示文档。
2. 实现了层次聚类算法，以及层次聚类算法和Ｋｍｅａｎｓ算法的结合。
3. 实现了Ｋｍｅｄｉｏｄ算法。

具体做法：

由于模拟器产生的坐标文件很多，每个文件包含1000个不同时刻的所有人的坐标观测值，所以在做聚类分析的时候是选取一定数量的文件，再从这些文件中选取某些记录来分析。

我的做法是程序每运行一次，选取20个文件，每个文件选10行（也就是10个时刻的观测值）。

可以看到，实际上，每一行观测值即某个时刻所有人的坐标已经足够做一次聚类，为什么要选取20\*10=200次的观测值呢？这是因为人群在商场中行走，有时会因为各种原因而走散，导致本来属于同一组的人在某个时刻的坐标相差很大，对聚类结果产生影响。

每次聚类选取一行坐标观测值，产生一次聚类结果。所以做聚类前要对数据进行预处理。

某一行的观测值 1 37.2 23.4 2 34.5 36.3 3 12.89 35.0 4

1 37.2 23.4

2 34.5 36.3

3 12.89 35.0

4 ....... ......

将数据变成这种格式后，才能做聚类。

如何处理这２００次（有２００行观测值）的结果？我用矩阵表示，每次聚类的结果用Ａ（ｋ）表示，ｋ＝１，２．．．２００，Ａ（ｋ）是一个行数=总人数=列数的矩阵，其中第ｉｊ个元表示在第ｋ次聚类中，第i 个人和第j个人有无在同一类，ａ（ｉｊ）＝１表示两份被分在同一组，ａ（ｉｊ）＝０表示没被分在同一组。Ａ（１）＋Ａ（２）＋．．．＋Ａ（２００）的和记为Ａ，显然Ａ的第ｉｊ个元表示在这２００个不同的时刻第ｉ个人和第ｊ个人有多少次被分在同一类，这个值越大，两人就越有可能在同一组。

由于选取文件和记录有任意性，所以我将程序运行多次，取平均值。程序运行结果将输出在ａｄｊａｃｅｎｃｙ．ｔｘｔ文件中，另外，还将输出ｇｒｏｕｐｉｄ．ｔｘｔ文件，

这个文件将记录每个人的真实预设组号，方便对比计算聚类的准确度。

（计算聚类准确度的算法由小组成员张泽宇实现。）

为了提高聚类的精确度，我尝试用不同的聚类算法处理每一个时刻的人的坐标数据。

就现在的结果来看，层次聚类的的精确度最高，Ｋｍｅｄｉｏｄ算法耗时最长，Ｋｍｅａｎｓ的精确度低于层次聚类，高于Ｋｍｅｄｉｏｄ，对Ｋｍｅａｎｓ算法的该进主要是优化初始凝聚点的选择，但同时也出现了“初始分配后出现空类”的问题，从而使聚类无法进行。

后续的任务仍然是优化已有的算法，并寻找不同的聚类算法，在软件中实现。

Ｒ文件：

1. ｐｏｓｉｔｉｏｎｃｌｕｓｔ１．ｒ：Ｋｍｅａｎｓ算法
2. ｐｏｓｉｔｉｏｎｃｌｕｓｔ２．ｒ：改进后的Ｋｍｅａｎｓ算法，选取尽量远的

　　　　点作为初始凝聚点。

３．Ｐｏｓｉｔｏｎｃｌｕｓｔ３．Ｒ：改进后的Ｋｍｅａｎｓ算法，用层次聚类的结

　　果作为Ｋｍｅａｎｓ初始凝聚点的参考依据。

1. Ｐｏｓｉｔｉｏｎｃｌｕｓｔ４.ｒ：层次聚类算法
2. Ｐｏｓｉｔｉｏｎｃｌｕｓｔ５．Ｒ：Ｋｍｅｄｉｏｄ算法

每个Ｒ文件都是独立的，不存在调用与被调用的关系，不同的Ｒ文件是不同的聚类算法

的程序。