我将给你一些伪代码，你需要根据这些伪代码写出python版本的代码

每台监测器具有七个数据，前三个分别是监测器的经纬度和高度，后四个分别是监测器接收到的四个音爆到达时间

第一步：数据准备

将七个监测器的经纬度转化为笛卡尔坐标

选出七个监测器中的前四个，将它们作为标准机，并将他们的空间坐标以及接受到的四个音爆的信号作为4行7列的矩阵存储

将七个监测器中的后三个作为验证机，将他们的空间坐标以及接收到的四个音爆的信号作为3行7列的矩阵存储

第二步：初始假设与求解

分别取出四个标准机中的每一个的某一个音爆到达时间，枚举所有可能的情况（256种），并对每种情况做最小二乘迭代求解对应的假设源的时空坐标。建立一个大小为256\*4的向量用于存储所有解。

注意，在这个过程中，我们需要记录每个解对应的选取的音爆到达时间的索引，以便我们后面调整数据。

第三步：验证与寻找真实答案

对于256种情况中的每一种枚举每一台验证机的四个音爆信号之一，计算音爆源的时空坐标和验证机音爆信号的匹配程度，每次枚举记录匹配程度最优的音爆信号。

这样我们就获得了对于每一种假设的源的位置，匹配程度最优的三个验证机音爆信号，对其索引记录之。

对于每一种假设的源的位置，计算整体的时间残差平方和，取时间残差平方和最小的源的位置为我们的解。

第四步：寻找剩下的三个音爆源的位置

将以上找到的第一个真实源对应的每台监测器的音爆信号从标准机和验证机的数据矩阵中去除，这样我们就拥有了4行6列的标准机矩阵和3行6列的验证机矩阵

重复与第二步、第三步类似的过程，找到第二个、第三个音爆源，当每台监测器只剩下一个音爆到达时间时，直接认为这是第四个音爆源对应的数据。

第五步：输出

输出四个音爆源对应的七个监测器收到的音爆到达时间，输出四个音爆源时空坐标的计算结果，输出四个音爆源对应的残差平方和的计算结果。