Problem 2:

思路：我们使用均匀随机采样的方法，选取s =100000,计算在不同的iteration对于计算的优化，一共进行了九次实验，在每次实验中，每次实验之间的s的选择是不同的。但在每组实验内考虑不同的iteration次数的时候，选择的s是相同的，这样可以排除随机性的影响，着重分析iteration次数对于计算精度的影响。  
之后选择i=10000，s =100000当作每组实验的真值，计算每组实验中不同迭代次数产生的相对误差。最后绘制9组实验在不同迭代次数处的平均相对误差。

观察到I =2000,相对误差小于0.1%。证明i取2000足够精度了，继续提高i的效率不足。

图形用户界面, 图表

描述已自动生成

图表, 折线图

描述已自动生成

之后，我们考虑在取I =2000 时，有多少的样本可以使得99%置信区间的半径小于1%预计值的大小，得出需要330000左右的数据点。

图表

描述已自动生成

图形用户界面

描述已自动生成

因此在后续的题目中，我们使用i= 2000， s = 330000计算三种采样方法的差距，这里的真值选则i=10000,s=1000000的随机采样。

文本

描述已自动生成

如何加速在一个MULTIPOLYGON的采样速度

inside\_mandelbrot, boundary\_points , outside\_mandelbrot= random\_sampling(current\_sample\_space,

                                                    num\_samples, max\_iter, iteration\_threshold)

文本

描述已自动生成

任务四要求您“制定并测试一种方法来进一步提高蒙特卡洛方法的收敛速度”。简而言之，您需要开发一种改进的蒙特卡洛模拟方法，以更有效地估算Mandelbrot集合的面积。具体步骤可能包括：

**研究当前方法的限制**：分析标准蒙特卡洛方法在估算Mandelbrot集合面积时的性能，包括收敛速度和准确性。

**探索改进策略**：研究和探索不同的技术和策略，如重要性采样、控制变量或使用改进的随机数生成方法，以减少方差和提高收敛速度。

**实施改进方法**：在蒙特卡洛模拟中应用这些改进策略，例如，通过选择更好的采样点或通过改变迭代的方式。

**测试和比较**：将改进后的方法与标准蒙特卡洛方法的结果进行比较，以评估收敛速度的提高和准确性的变化。

**文档化改进**：记录您所采取的改进措施，包括任何新策略的理论基础，以及这些改进如何影响了收敛速度和结果的准确性。

**性能评估**：通过重复试验，对改进方法的性能进行全面评估，并与原始方法进行对比。

为了开始这个过程，您可以选择下面的某些方法：

**重要性采样**：通过更频繁地采样那些对最终结果影响更大的区域来提高效率。

**分层采样**：将复平面分成多个区域，并确保每个区域都被均匀采样，以减少方差。

**拉丁超立方采样**：确保在每个维度的投影上均匀采样。

**正交采样**：在样本点之间引入正交性，以减少样本点的相关性。

**自适应方法**：根据已经计算出的点的特征动态调整采样策略。

在您开始编码之前，应该先深入理解这些方法，并决定哪一种或哪几种结合起来可能最有效。您可能还需要阅读相关的文献或查找资料来获得更多关于改进蒙特卡洛模拟的想法。

t

thomas,关于part4, 我的思路是这样的，确定迭代上线为2000次，x\_range = [-2, 1]

y\_range = [-1.5, 1.5]

首先使用10000点进行随机采样，根据每个点最终的迭代次数i，当5<i<2000,将这些点构建成一个边界区域，这些个边界应当是联通的，对于区域内部的点，我们认为这是必定属于Mandelbrot集合，区域外部必定不属于Mandelbrot集合。（详见我的代码yunalu.ipynb的q4部分）

之后对于边界区域进行10000点进行随机采样，确定边界区域中属于Mandelbrot集合的概率，重复以上操作几次，得到高精度的结果。

这里每次采样的方法，采样数，以及边界区域的确立，a<i<2000（迭代下限a的选择）都是可以调整的。

Run1:

文本

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

Run2:

文本

描述已自动生成

图形用户界面

描述已自动生成