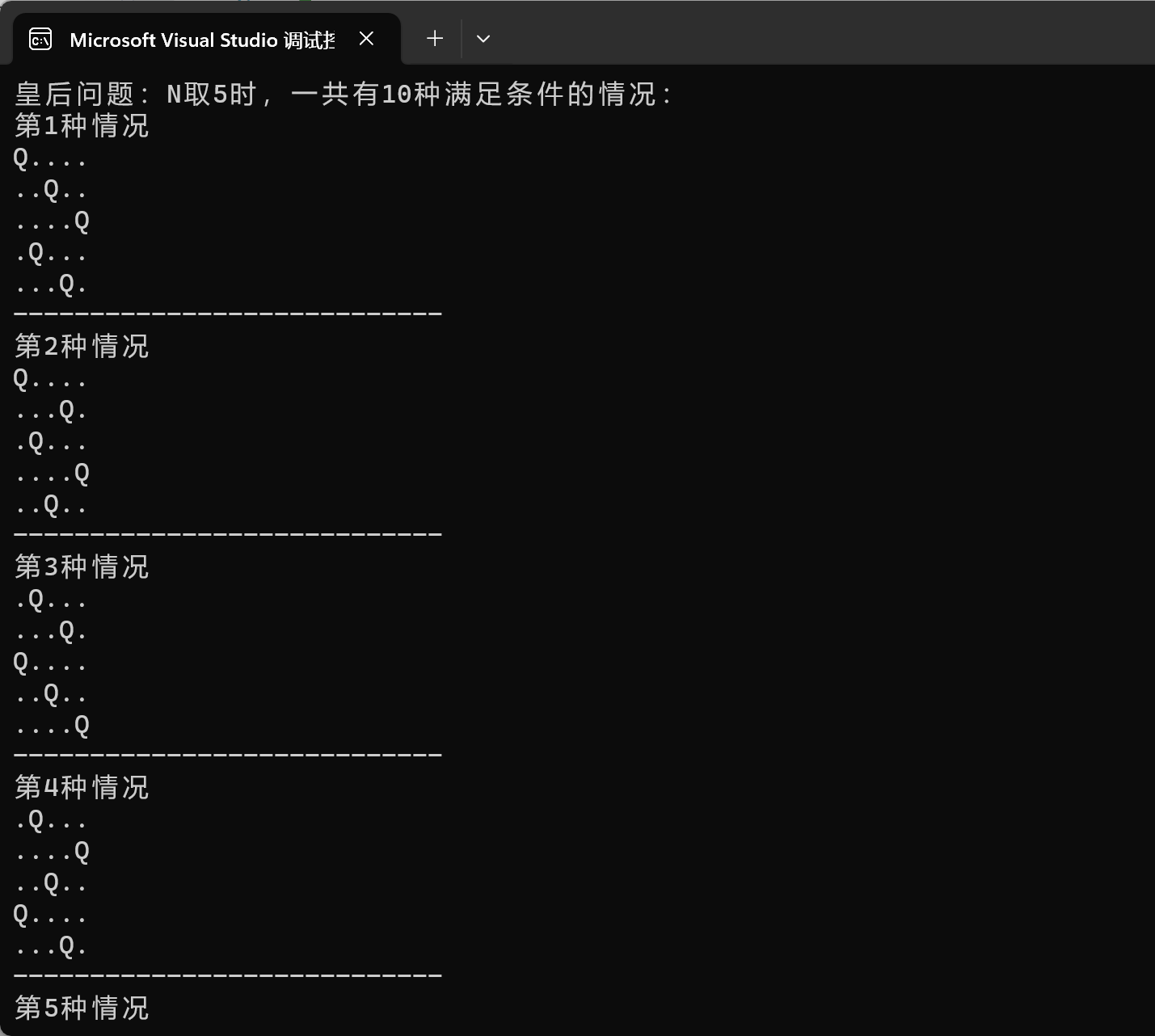
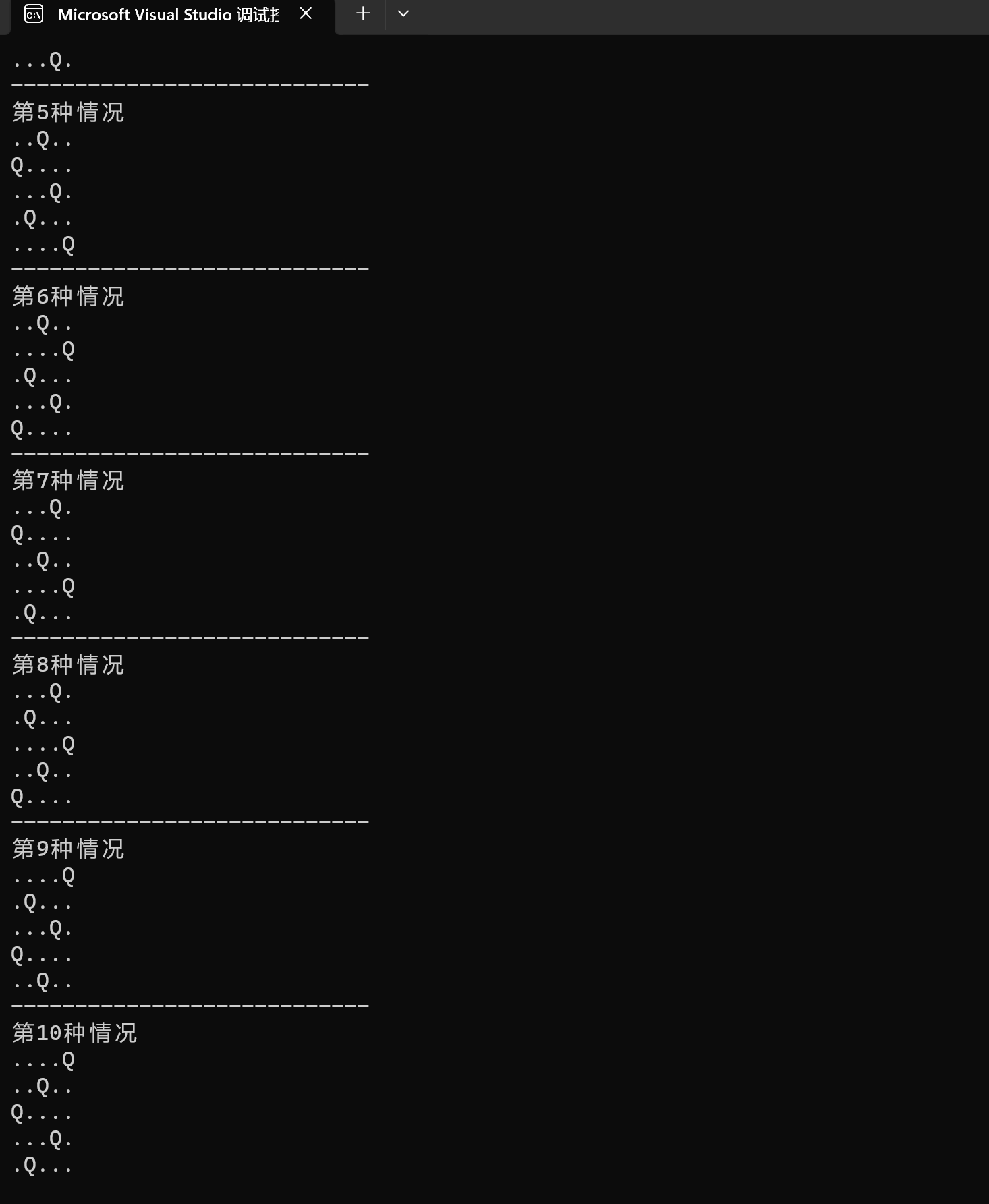
1. 皇后问题

这里运用回溯解决皇后问题：回溯法是一种通过尝试所有可能的候选解并在搜索过程中寻找可行解的方法。在解决八皇后问题时，这种方法首先从棋盘的第一行开始，在每一列中选择一个位置放置皇后，并通过递归处理下一行。在每个决策点，通过检查当前位置是否和已放置的皇后产生冲突来判断选择是否有效。如果当前选择导致冲突，就回溯到上一个决策点，重新选择。当成功放置了所有的皇后时，找到了一个可行解。这个过程一直进行，直到搜索完所有可能的放置方式。

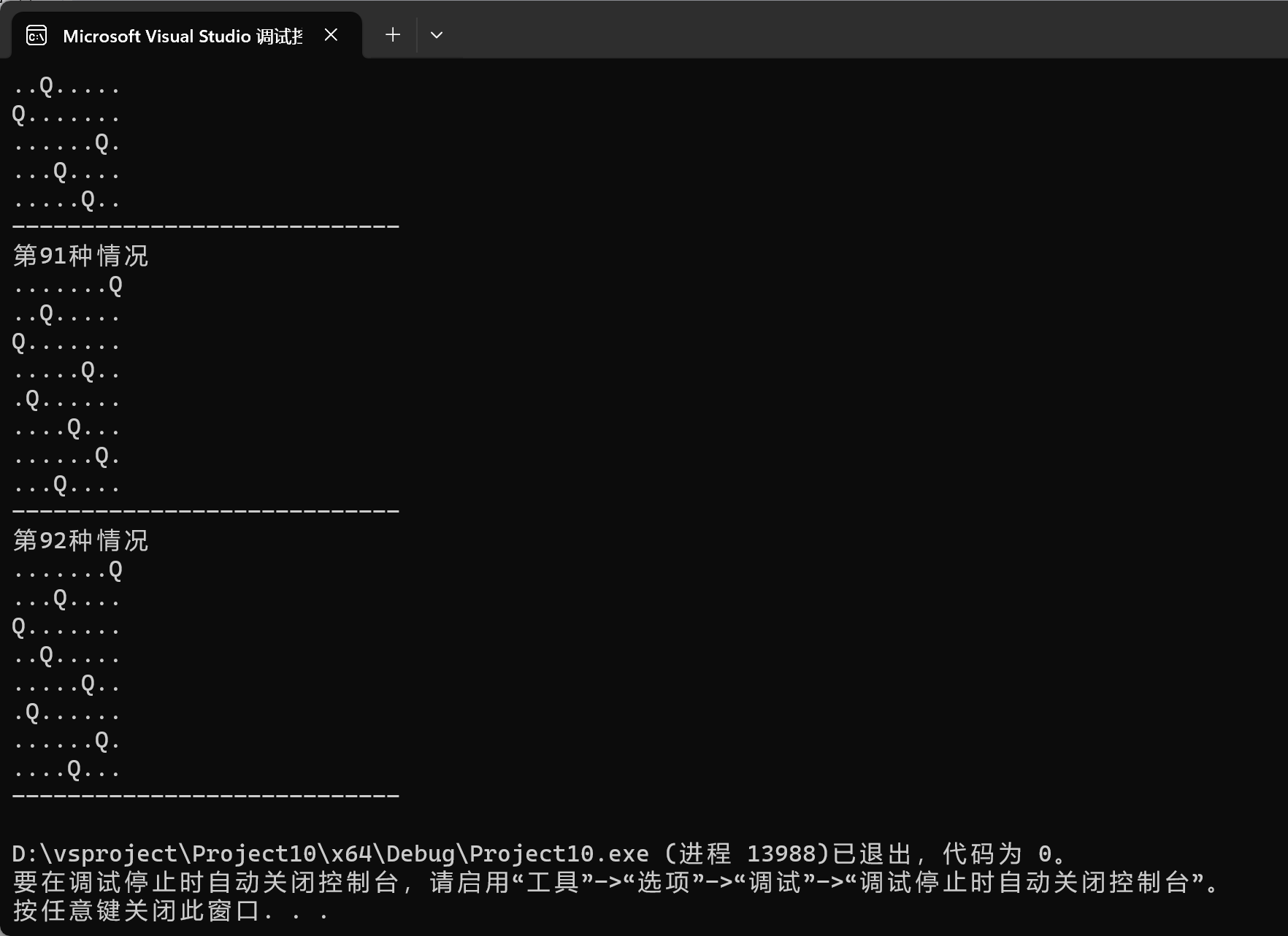
N=5一共有十种情形，**.** 代表空位,Q代表皇后的位置：





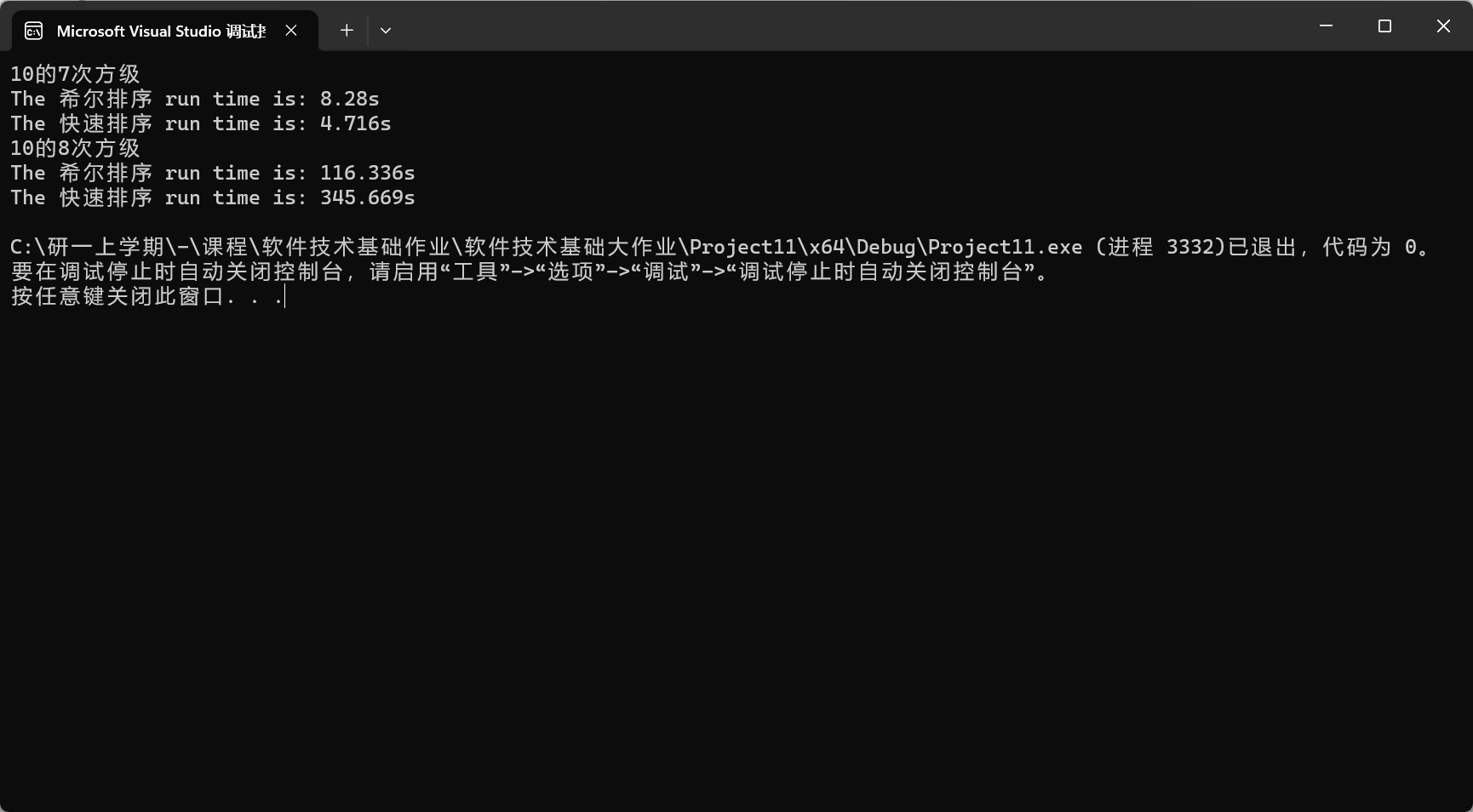
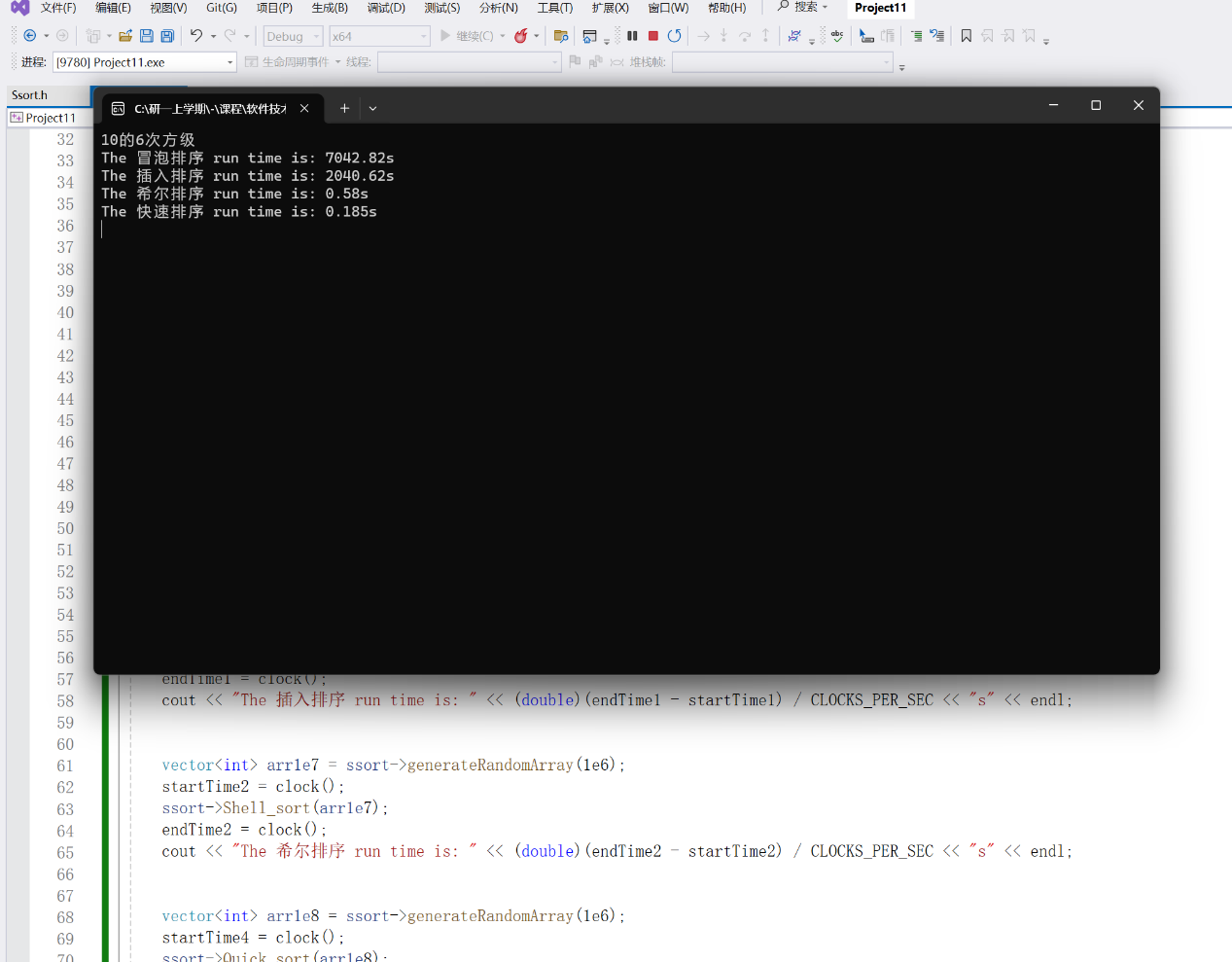
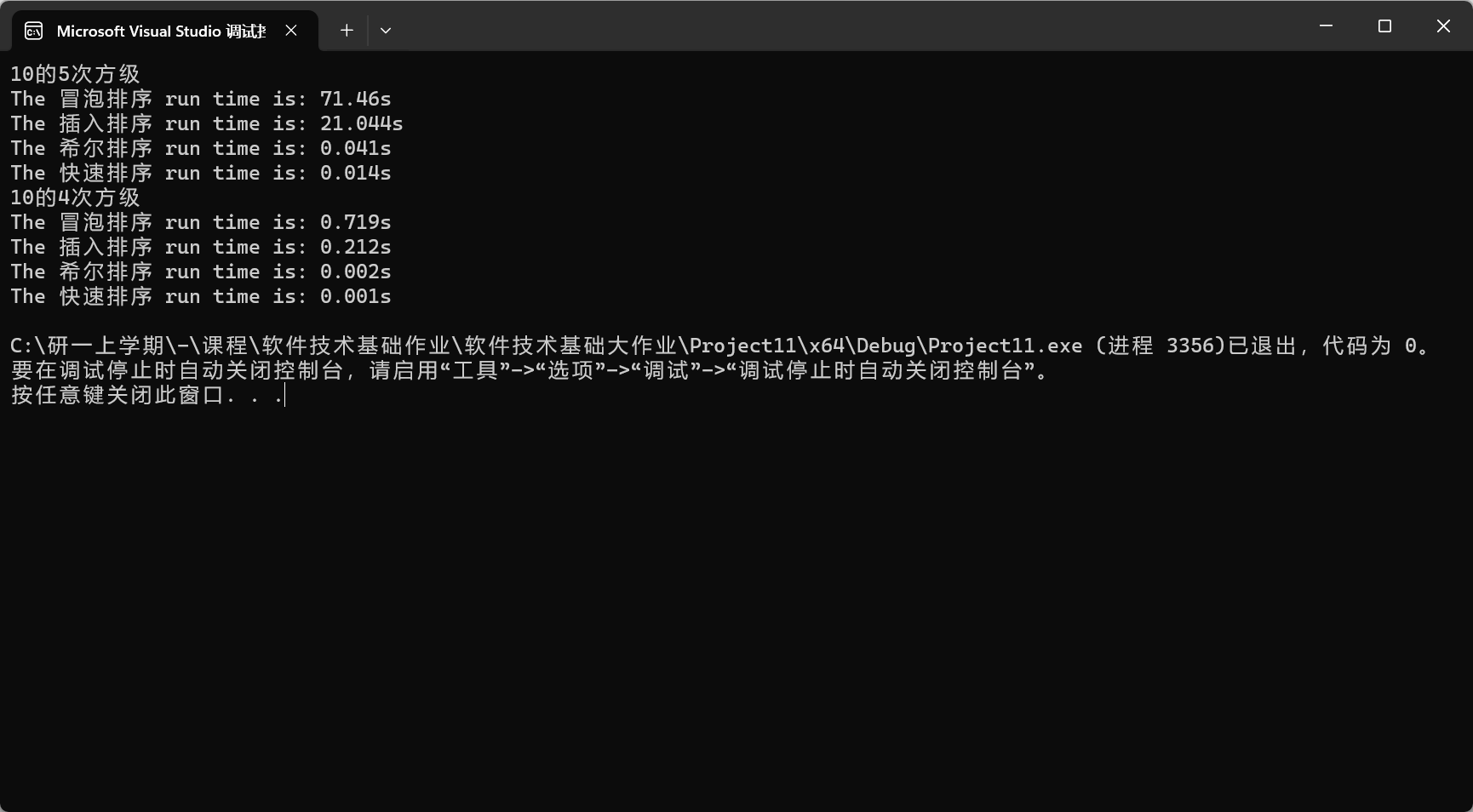
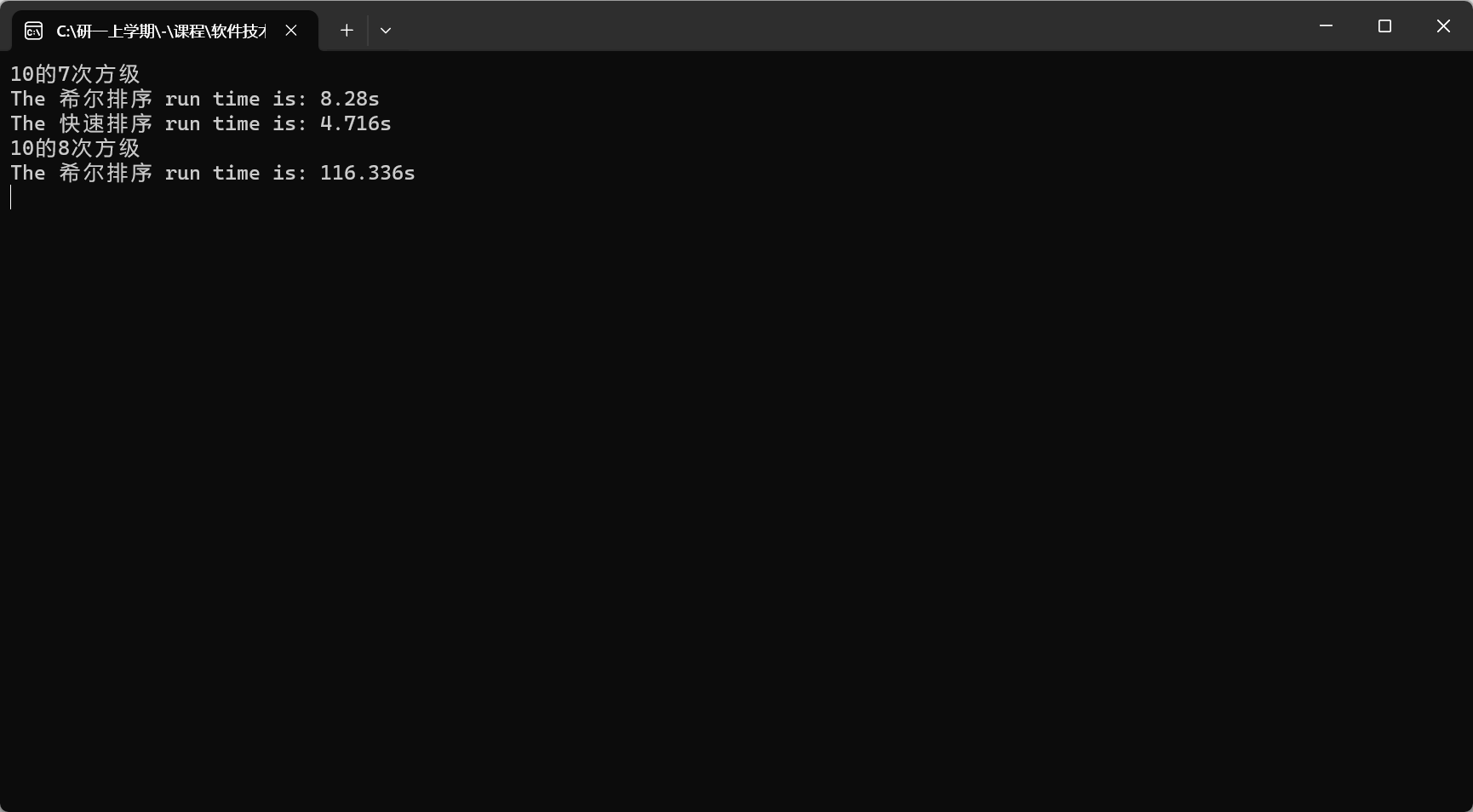
当N=8时，实验结果如下，一共有92种解法，中间部分不再列出：





2、排序问题

这里选择四种排序算法，分别是插入排序、冒泡排序、希尔排序、快速排序，这里选择了1e4、1e5、1e6、1e7、1e8个数据上的排序速度，以下是排序结果：



在本实验中，冒泡排序、插入排序的时间在1e6就已经达到了7000s，因此在1e7和1e8数量级中没有使用冒泡排序和插入排序。

在一般情况下，快速排序（QuickSort）通常比希尔排序（Shell Sort）更快，尤其是在大量数据上。希尔排序的性能通常介于冒泡排序和快速排序之间，而快速排序在平均情况下具有较好的性能。

希尔排序的时间复杂度取决于选择的间隔序列，但在最坏的情况下，它可以达到O(n^2)。快速排序的平均时间复杂度为O(n log n)，它在实践中通常表现得相当快。

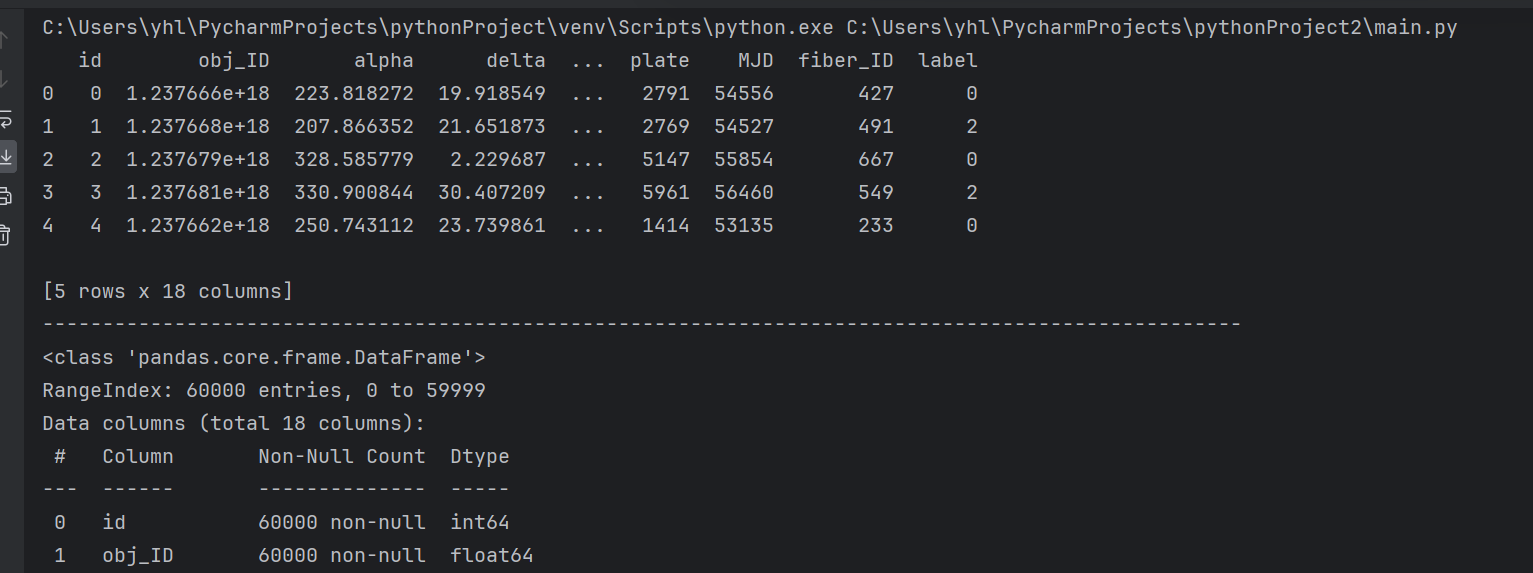
在平均情况下，快速排序的时间复杂度为O(n log n)，而希尔排序的时间复杂度通常介于O(n log^2 n)和O(n^2)之间，具体取决于所选择的间隔序列。快速排序的较低时间复杂度使其在大规模数据集上表现更好。

快速排序在排序过程中通常涉及到相邻元素之间的比较和交换，这符合计算机体系结构的局部性原理。相比之下，希尔排序的交换跨越较大的间隔，可能导致内存访问的散乱性，降低了缓存的命中率。

作业三：恒星数据集分类问题

这里选择了一种提升树的方法（决策树），GBoost的分布式版本有广泛的可移植性，支持在Kubernetes、Hadoop、SGE、MPI、 Dask等各个分布式环境上运行，使得它可以很好地解决工业界大规模数据的问题。

1、导入数据：



2、数据清洗

1. 删除无效字段，如从字段含义和数据分布可以观察出哪些字段对于模型训练没有帮助

2. 缺失值处理：将缺失值过多(如占比超过30%)的字段删除，将缺失值较少(占比少于30%)的字段用中位数、平均值、随机值进行填充

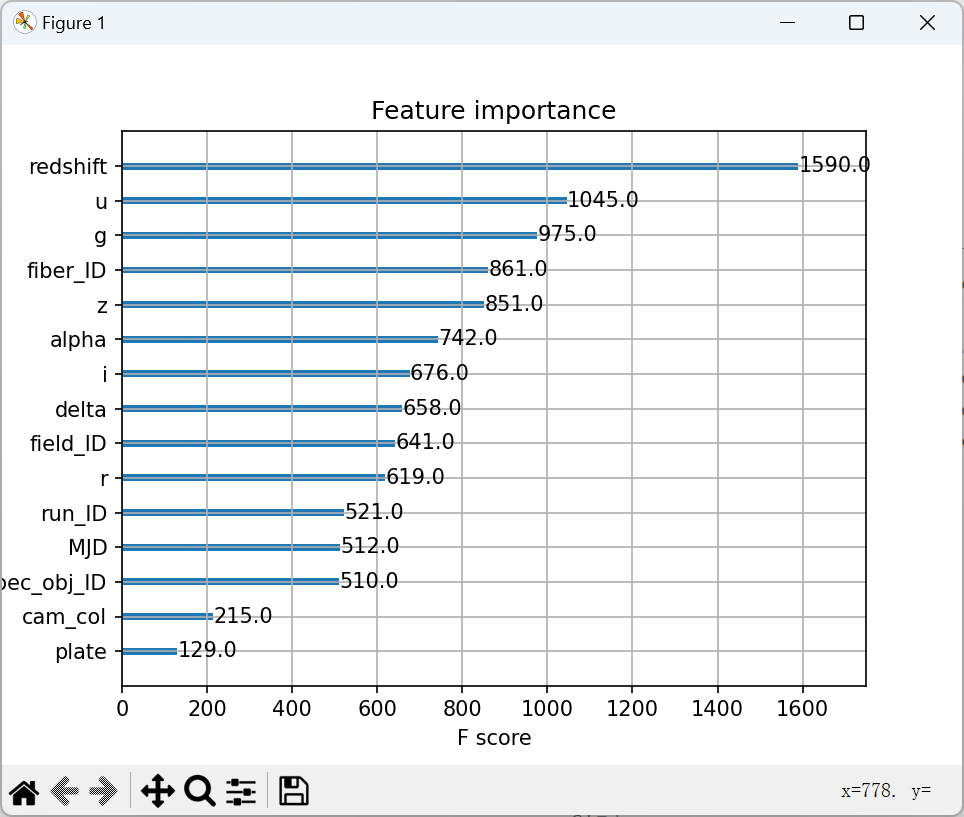
3.发现给中标签值下的数据数量不均衡，因此将所有类型数据的数量调成相同值，这样就可以使训练更加均衡，数据集中的数据类型大致为3：1：2.

3、特征选择

1. 利用任务二得到的数据对模型进行训练

2. 利用训练好的模型获取其feature\_importance参数值

3. 选取排名靠前的部分字段作为新的训练数据



选取排名前几的特征，进行数据的分类特征

4、训练评估

将训练集数据7：3划分成训练数据与测试数据，在训练集上由98.3%的正确率。将训练好的模型对所给的测试数据进行预测，完成实验。

