

# LAB1

## 实验名称：比较元素选择算法

### 实验内容：

1. 随机数据集的生成。数据集的生成采用 c++ 语言，与后续元素选择使用的语言一致以便于调试。考虑到 `std::rand()` 生成的数字可能不足以覆盖所有范围，我们选择了 `random` 库提供的 `std::mt19937` 随机数生成器，并把 `std::time(0)` 作为随机数种子，最大值设置为数据集的 10 倍。
2. 元素选择算法实现。归并排序，即先使用归并排序将数据排序，再输出中位数。具体的算法实现在目录的 `Divide_Merge.cpp` 中。线性时间选择，分为含五个元素的小组做剪枝来降低时间复杂度，具体实现在 `linear_time_select.cpp` 中。`nth_element()`，我们选择直接调用 STL 中的算法运行，具体实现位于 `kth_element.cpp` 中。上述三种算法均由 c++ 实现。
3. 计时方法。我们选择调用 `chrono` 库的 `high_resolution_clock` 实现微秒级的计时。其中数据集保存在文件中，没有将输入的时间计入。
4. 编译及运行环境。在 Windows11 + clion + MinGW11.0 + cmake3.27 + Debug 环境下完成编译(课上提到 Release 时已经完成了实验)。运行在 Windows 10.0.22621.2506 + 13th Inter i5-13500H(16 CPUs) 2.6GHZ + 16G RAM + LNVNB161216 + NVIDIA RTX 4050 环境下。
5. 图片结果输出。三种算法在每个数据集的时间通过 python 的 matplotlib 作图，图片保存在目录的 `image` 文件夹中。最大数据集下运行实例的截图与作图如下：

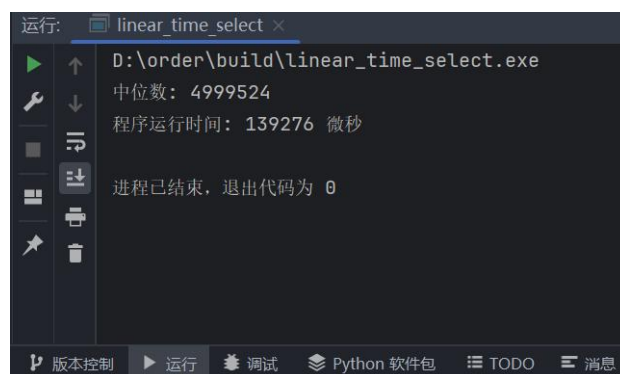


图 1 线性时间选择

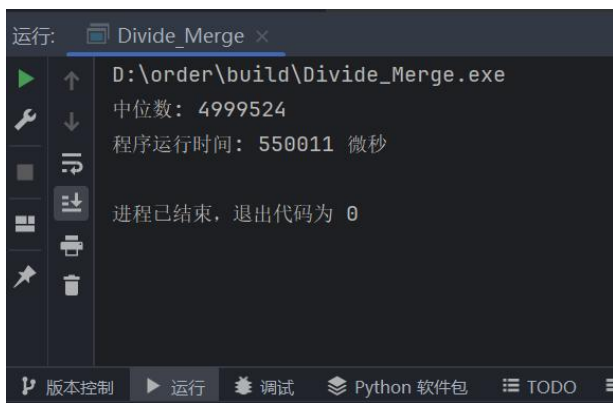


图 2 归并算法

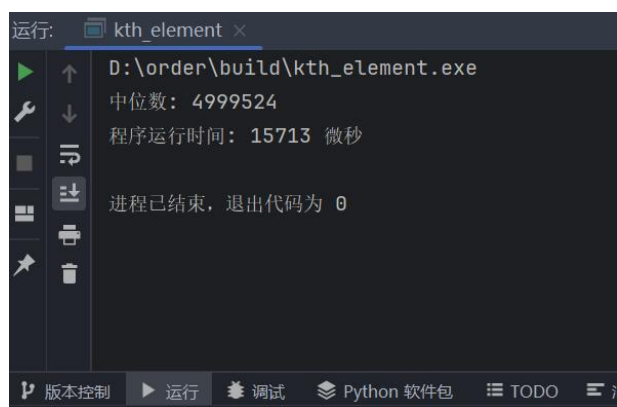


图 3 nth\_element()

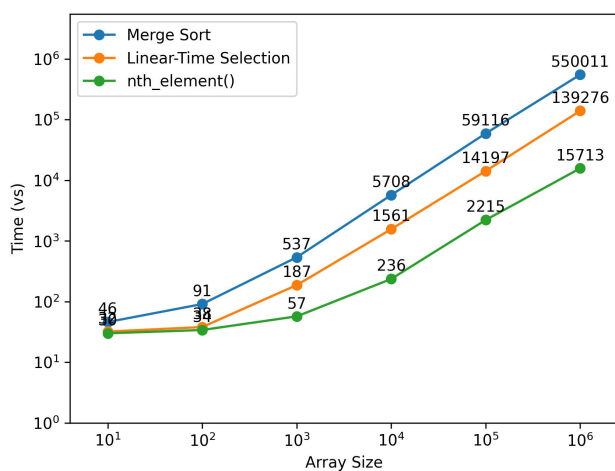


图 4 T-Size 折线图

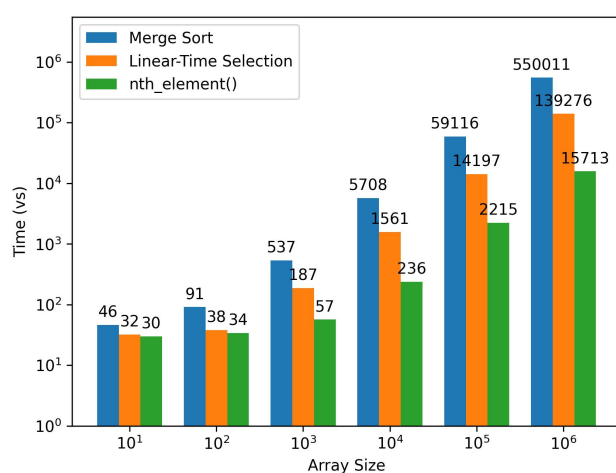


图 5 T-Size 柱状图