

运行环境要求:

能够支持 g++ 编译的操作系统即可.

程序运行:

所有代码都在 segBitonicSort.cpp 文件中, 只需要在命令行运行

```
g++ segBitonicSort.cpp -o sort -lpthread
```

就可以生成可执行文件。

Windows 下执行 sort.exe

Linux 类系统下执行 ./sort

即可, 输入输出测例已经在代码内部写好了

程序输入输出说明:

程序没有外部输入, 代码中随机生成了 10000000 个数据, 保存在 data.txt 文件中, 作为排序算法的输入。

程序在单线程和 10 个线程下 (电脑最多支持 12 线程) 分别进行分段双调排序, 分别会输出 output_single.txt, output_mt.txt 以及 result.txt 三个文件。output_single.txt 文件是在单线程情况下排序后的输出文件, output_mt.txt 文件中是在 10 个线程下排序后的输出文件, 这两个文件的内容是完全一样的。result.txt 中存放了两种不同情况下运行时间以及耗时对比。

附加说明:

1. 在试验报告中给出了更多中并行排序情况的对比, 这里只给出了一种可运行的配置, 方便检查程序
2. 算法对存储, 内存以及 CPU 占用比较高, 在 10000000 个数据排序配置下, 输出文件就会消耗 2GB 左右存储空间。如果测试时自行设置参数的话, 不要设置太高。
3. 排序理论算法时间复杂度是 $O(n \cdot \log n \cdot \log n)$, 经过我自己的计算和验证, 算法满足理论复杂度计算出的时间。
4. 实验的对比结果在 result.txt 文件中, 基本格式如下:

```
1 Sort with single Thread spent: 2.410 sec
2 Sort with 5 Thread spent: 1.394 sec
3 Sort with mt_threads is 1.72947 times faster than single thread.
```