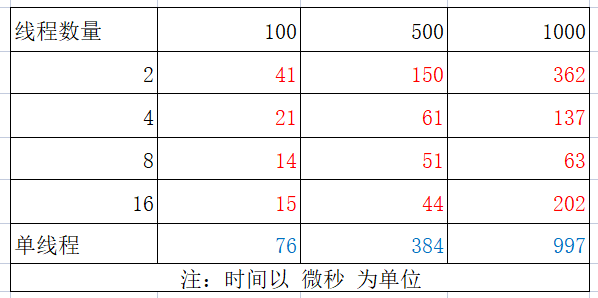
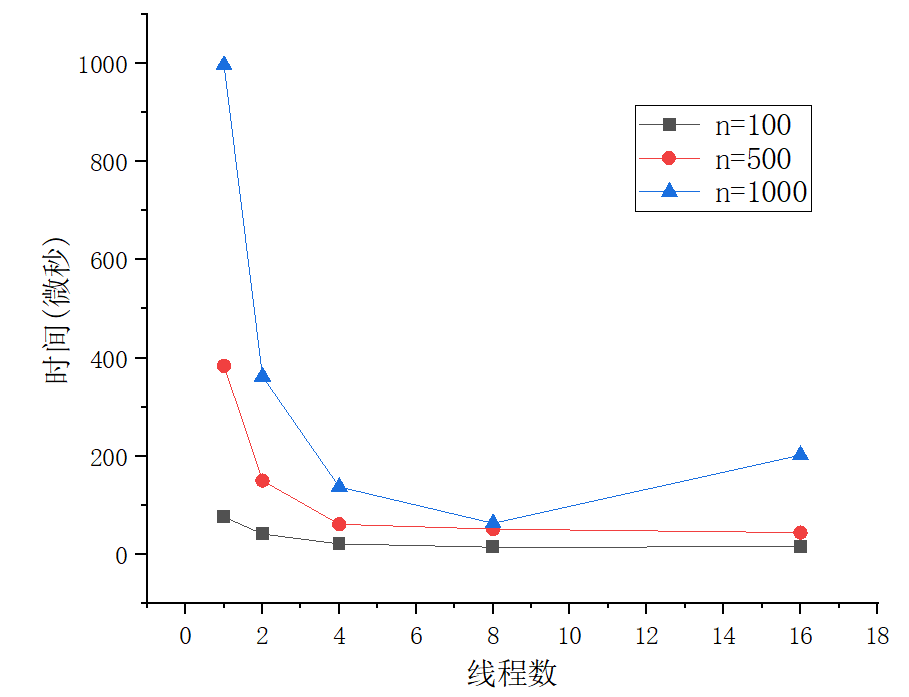
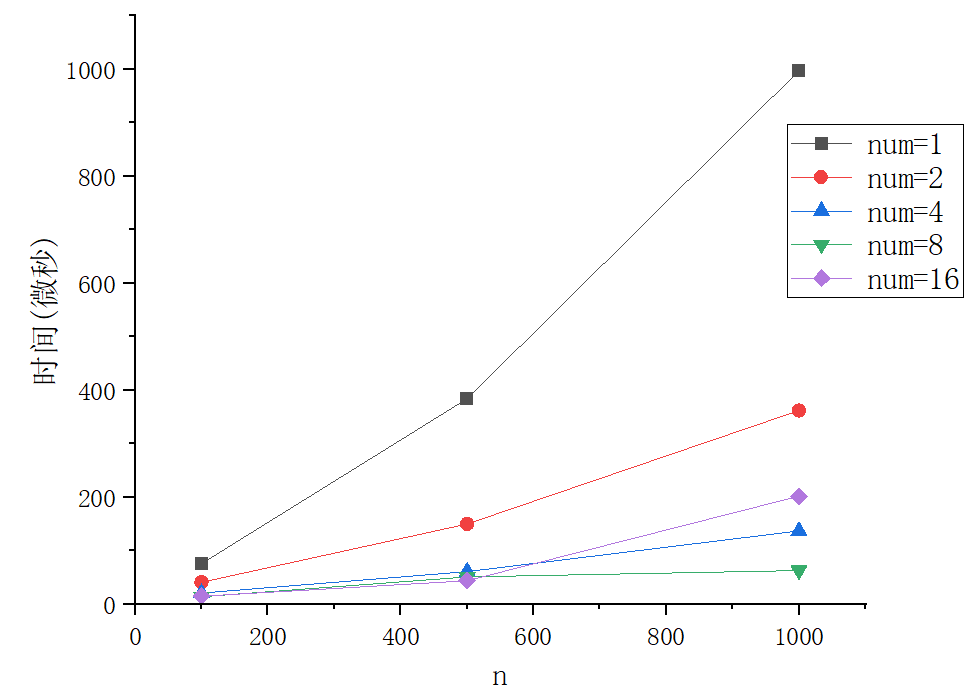
1. 由于程序的并行执行，可能会出现某个线程的printer函数还未执行完，却被其他的线程回收掉，导致出错。
2. 会导致死锁，可能出现死锁的位置是：thread1在进行a.lock()之后，thread2正好运行完b.lock(),之后thread1与thread2均会阻塞在各自代码的对应的下一行，即此时ab两把所全部被锁住，无法使用，导致死锁，但是这只是一种情况，并不一定每次都发生。
3. 首先计算总的容器长度，之后根据分配的线程数量N来计算每一部分的size，之后在每一个piece上面分配一个线程，并使这个线程运行for\_each函数的对应部分，之后在全部运行结束之后进行线程的回收。
4. 解题思路：

首先根据threadNum将原始数组分成大小相等的blocks，之后利用循环创建多个线程，使得每个线程执行串行的mergesort算法来对不同的block进行归并排序，之后使用多路归并算法来对每个排序后的block进行总体的归并，并得到最终的结果。







结论：从上面的图表可以看出，对于相同的线程数，归并排序的时间几乎成线性增长，而对于相同的数组规模而言，在线程数量为1--8时，性能随着线程数量的增加而线性提升，而线程数量为16的时候性能会出现平衡或者下降的情况，这可能是由于测试是泡在最多同时运行8个thread的机器上面，所以thread为16个时会出现由于较大的竞争而导致的overhead，这就可能导致性能下降。