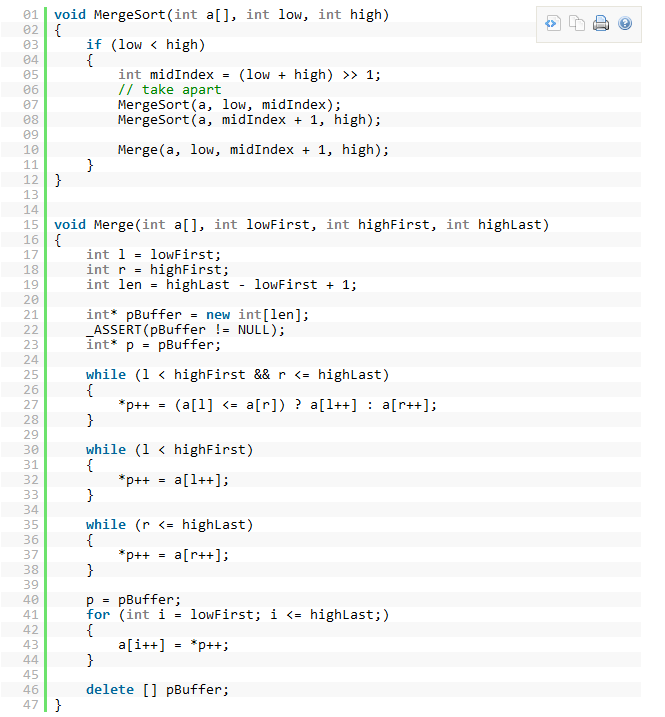
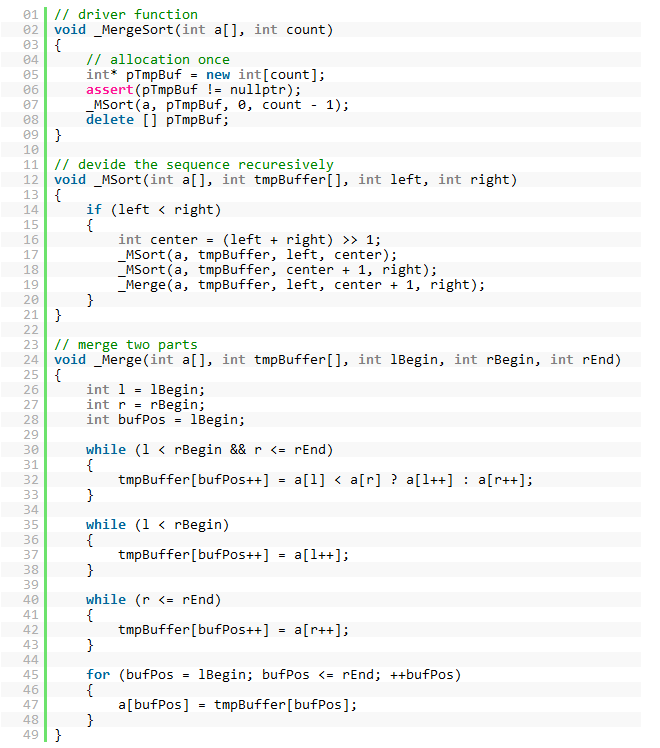
**更改内存分配策略改善归并排序效率**

归并排序是一种相当稳健的排序算法，无论何种输入序列，其期望时间复杂度和最坏时间复杂度都是Θ(nlogn),这已经达到了基于比较排序算法的渐进下界。因此归并排序时常会用于对可能导致quicksort退化的序列排序。

归并排序是典型的分治算法，一个最常见的实现如下：

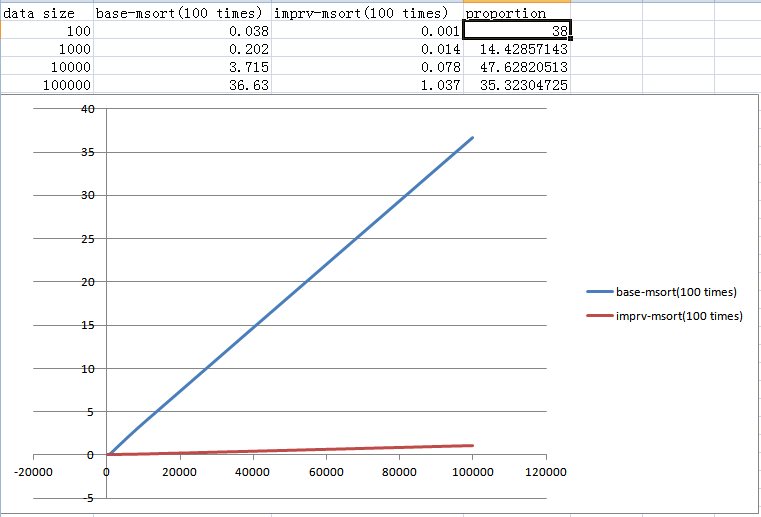


但是在实践中，归并排序花费的时间往往超过预期，对于普通的序列而言，所花费的时间甚至远远超过quicksort。究其原因，和归并排序的内存策略有关。归并排序不是原地排序，需要额外的存储空间。并且在每次Merge过程中，需要动态分配一块内存以完成对两个数据堆的排序合并。并且排序完毕之后，我们需要将存储空间中的数据复制并覆盖原序列。最后一步操作是由归并排序自身性质决定，无法优化，所以我们只能针对Merge操作。经过分析很容易知道，对于长度为n的序列，要执行logn次的merge操作，这意味着需要进行logn次的内存分配和回收。内存操作开销较大。  
如果能够一次性分配长度为n的存储空间，那么就省掉了大量的分配操作，可以极大提高效率。由于归并的分治特性，我们需要在原来的函数基础之上，包装一层驱动函数(driver function)



为了检验性能提升，我对100，1000，10000，100000的数据规模分别进行测试，每次测试排序运行100次，得到如下数据表和图示。

ps：为了减少干扰，以上测试均在Release下进行。

[](https://byfiles.storage.live.com/y1pHF4WjhSZUTN09iFrZBJvT0XoyeOqWnZwv1XkZ7zTpyN7dJ-sel9LFZ2YICInM5r3fEh4ToifYWI/datacollection.PNG)

经过测试发现，改进后的算法基本上比原始的要快30～50倍。至于表格中1000测试那行，猜测和缓存命中有关。多次测试后，基本也快出30-40倍。

**pps：至于那个折线图，如果不对请无视，不怎么会用Excel画图表。。**

以上改进可以得到一个结论：对于需要频繁分配内存的算法而言，一次性分配或者采用lazy-deletion以提高复用的策略可以大幅提高算法效率。