1 设计思路

核心并不复杂，只有 HeapSort 一个函数。先创建空堆，此时数据已经存储在 HeapSort 中的 vc（也就是 test.cpp 中的 v），从第一个元素开始一个一个插入堆，而“插入”在代码中具体体现为“将 push_heap 的两个参数中的后一个 +1”，相当于文献中的 vc.push_back()，再进行 push_heap。所有数据遍历后，堆就创建完成了，默认为大根堆。

每次将最大值（也就是 root）进行 pop_heap，这一函数具体做了“将 root 与堆的最后一个调换位置，此时的最后一个元素不算作属于堆的了，然后调整剩余元素使之仍为堆”，每次将 pop_heap 的后一个参数-1，代表这个是已经排序好的，不再算作属于堆的了。由此从后往前排序。

2 测试流程

共 4 组，分别为随机序列、有序序列、逆序序列、部分元素重复序列，其中部分元素重复序列为生成 0, 1, ..., int(size / 100) - 1 如此循环的序列。每组先按要求生成数据存到 vector v 中，然后分别进行 HeapSort、检查排序正确性、使用 STL 的 HeapSort、清除数据。

3 测试结果

每次测试结果有误差（±30% 以内），这里取其中一组测试结果。

	my heapsort time(ms)	**std::sort_heap time(ms)**
random sequence	172	80
ordered sequence	92	88
reverse sequence	78	89
repetitive sequence	126	65