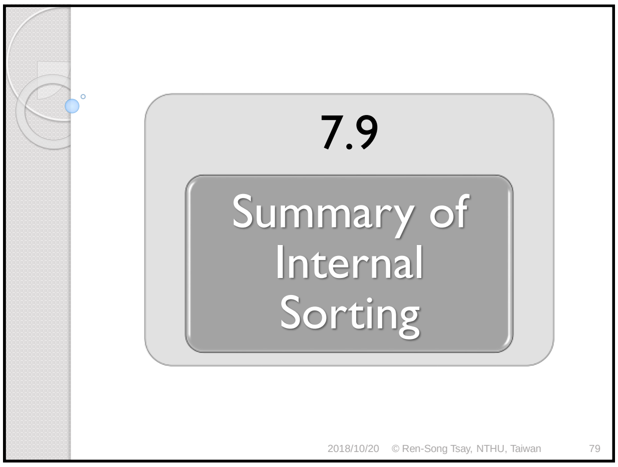
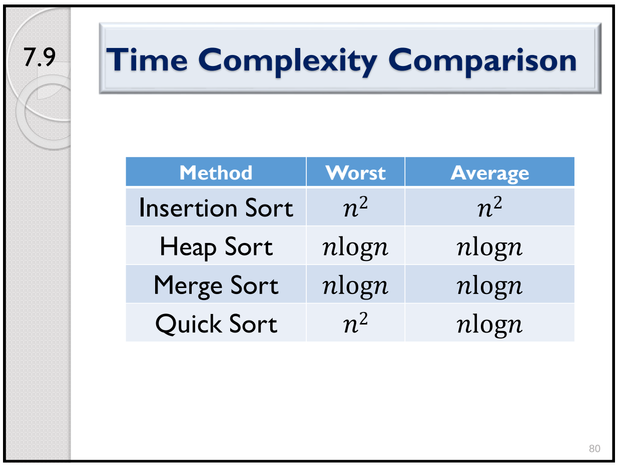


7.9

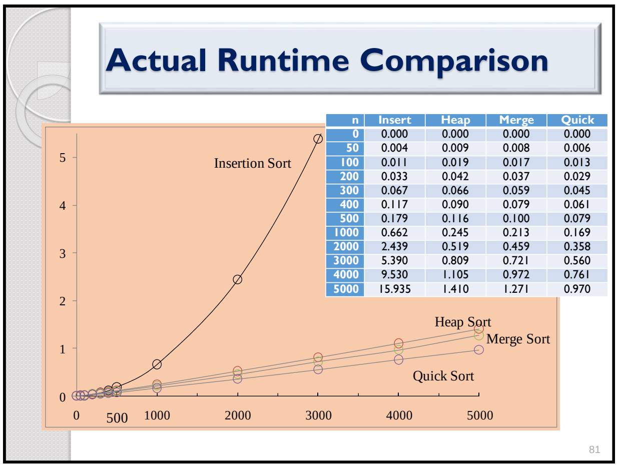
Summary of Internal Sorting



7.9

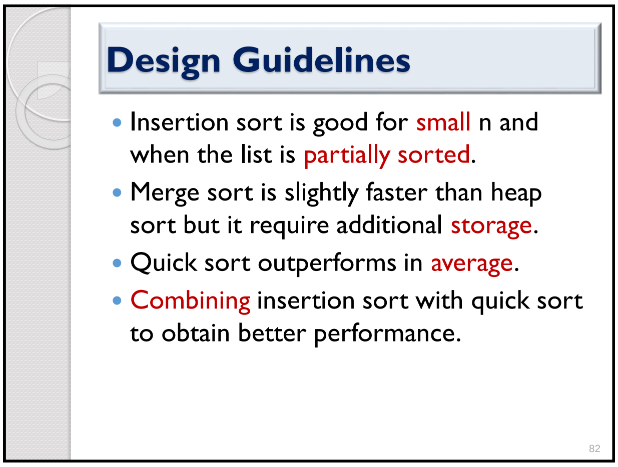
Time Complexity Comparison

Method	Worst	Average
Insertion Sort	n^2	n^2
Heap Sort	$nlogn$	$nlogn$
Merge Sort	$nlogn$	$nlogn$
Quick Sort	n^2	$nlogn$



Actual Runtime Comparison

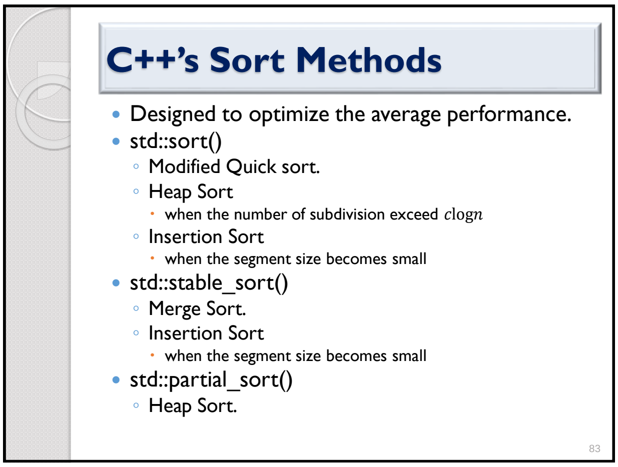
n	Insert	Heap	Merge	Quick
0	0.000	0.000	0.000	0.000
50	0.004	0.009	0.008	0.006
100	0.011	0.019	0.017	0.013
200	0.033	0.042	0.037	0.029
300	0.067	0.066	0.059	0.045
400	0.117	0.090	0.079	0.061
500	0.179	0.116	0.100	0.079
1000	0.662	0.245	0.213	0.169
2000	2.439	0.519	0.459	0.358
3000	5.390	0.809	0.721	0.560
4000	9.530	1.105	0.972	0.761
5000	15.935	1.410	1.271	0.970



Design Guidelines

- Insertion sort is good for small n and when the list is partially sorted.
- Merge sort is slightly faster than heap sort but it require additional storage.
- Quick sort outperforms in average.
- Combining insertion sort with quick sort to obtain better performance.

82



C++'s Sort Methods

- Designed to optimize the average performance.
- std::sort()
  - Modified Quick sort.
  - Heap Sort
    - when the number of subdivision exceed $c\log n$
  - Insertion Sort
    - when the segment size becomes small
- std::stable_sort()
  - Merge Sort.
  - Insertion Sort
    - when the segment size becomes small
- std::partial_sort()
  - Heap Sort.

83