Data Structure Project 1

Deadline: Apr 28, 23:59

This project requires students to compare five sorting algorithms, which are "Bubble Sort", "Insertion Sort", "Merge Sort", "Quick Sort", and "Heap Sort" in the aspect of time complexity, best&worst case scenario.

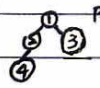
Requirement:

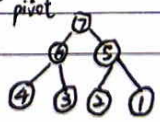
1. Implement the five sorting algorithms based on the skeleton code provided.
2. Compare the running time of five sorting algorithms, and fill the following table:

t	3	4	5	6	...	14	15	16	17
Bubble	0	0	0	0		0.81	3.336	14.56	42.821
Insertion	0	0	0	0		0.245	0.814	2.85	9.924
Merge	0	0	0	0		0.018	0.026	0.084	0.153
Quick	0	0	0	0		0.009	0.02	0.051	0.13
Heap	0	0	0	0		0.002	0.008	0.02	0.03

where each cell in the table denotes the running time (recorded by C++ timer) given the input size (number of elements in the list to be sorted) 2^t. For example, at column "17", each soring algorithm should sort the list containing 2^{17} random integers. Note: in order to be fairness to all the sorting algorithms, the input random integer list should be the same.

3. Use "t" as X-axis and running time (value in each cell in above table) as Y-axis, plot all the points and sketch the curve (You may do this by Excel) for each sorting algorithms. Draw all five curves in one X-Y coordinate plane. Compare the five curves and explain the reason.
4. Describe the best/worst case and the corresponding time complexity of each sorting algorithm. You may fill the tables below:

	Best case description	Best case example	Best case time complexity
Bubble	sorted in increasing order	1 2 3 4 5	$O(N)$
Insertion	sorted in increasing order	1 2 3 4 5	$O(N)$
Merge	no need to mergesort	(1 2 3), (4 5 6)	$O(N\log N)$
Quick	partition is perfectly balanced pivot is always in middle	1 2 3 4 5 pivot	$O(N\log N)$
Heap	least nodes sorted in order		$O(N\log N)$

	Worst case description	Worst case example	Worst case time complexity
Bubble	sorted in reverse order	5 4 3 2 1	$O(N^2)$
Insertion	sorted in reverse order	5 4 3 2 1	$O(N^2)$
Merge	mergesort to sort n numbers	(2 4 6), (1 3 5)	$O(N\log N)$
Quick	The pivot is the smallest element	1 2 3 4 5 pivot	$O(N^2)$
Heap	most nodes sorted in a reverse order		$O(N\log N)$