1. Random sort (Kind of sorting \ Number of numbers)

	5	50	100	1000
Bubble sort	0.001068	0.006728	0.195888	0.729689
Insertion sort	0.000653	0.005749	0.011966	0.225484
Selection sort	0.000855	0.005754	0.012297	0.251864

2. Nearly sort (Kind of sorting \ Number of numbers)

	5	50	100	1000
Bubble sort	0.001458	0.000861	0.002722	0.496757
Insertion sort	0.000148	0.000343	0.000620	0.018538
Selection sort	0.000157	0.007190	0.001786	0.153393

Write a paragraph discussing the results in terms of complexity and other characteristics of the three sorts.

Explain about sorting and time complexity.

1. Bubble Sort

Bubble sort is comparing with next elements and if they want to sorting with ascending way and when next element[n+1] is smaller than current element[n], it will swap position.

And this way will keep continuing until when it compares with last element.

Through the for loop and another for loop which is inside the previous for loop, it will be run.

When It comes to time complexity, if array is almost sorted, time complexity would be O(N).
But when array is not sorted like Reversely sorted, time complexity would be O(N^2).

2. Insertion Sort

Insertion sort starts with second index element. Current element will be saved in some variable and comparing element will be (current element -1).

Current element will keep comparing with all other element which is on the left side.

And When it's smaller than comparing element, it will change the position.

When It comes to time complexity, if array is almost sorted, time complexity would be O(N).
But when array is not sorted like Reversely sorted, then time complexity would be O(N^2).

3. Selection Sort

Selection sort is divided into two parts, the sorted part at the left end and the unsorted part at the right end.

The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. And this process continues moving unsorted array boundary by one element to the right.

However the array is, selection sort will compare whole elements, so time complexity will be O(N^2).

Explain about characteristics of three sorts.

1. Bubble Sort

Bubble sort takes more longer time than other sorting. I guess it shouldn't really be used in real life unless when teacher shows the way of how bubble sorting is or unless the data is already almost sorted in the scenario when the item is not really far away from sorted position.

2. Insertion Sort

Insertion Sort is pretty good on sorted array when the N is small and big. When you know the array is nearly sorted or if it's small length of array (N<50), insertion sort is good way to go.

Because when N is bigger than 50, the time is getting similar with selection sort. That's why when N is smaller than 50, Insertion sort is better than selection sort.

3. Selection Sort

Selection Sort is effective for very short length.

But when the array is nearly sorted, it is not a good idea because it will compare all elements comparing with insertion sort.

It looks like insertion sort is better than selection sort, but when it comes to number of swaps, when selection sort is O(n), the insertion sort is O(n^2). It will be important for flash memory because it will reduce the lifespan of flash memory.