|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **100** | **1k** | **10k** | **50k** |
| Linear Search | 37 | 312 | 3337 | 13146 |
| Bubble Sort + BS | 23183 | 23148 | 23643 | 23182 |
| Selection Sort + BS | 6185 | 5976 | 6076 | 6150 |
| Insertion Sort + BS | 9921 | 9270 | 9397 | 9466 |
| Merge Sort + BS | 78 | 84 | 93 | 159 |
| Quick Sort + BS | 63 | 52 | 60 | 109 |
| Radix Sort + BS | 85 | 54 | 66 | 133 |

Time-running table (in millisecond)

In case of tiny-size-data, Linear Search takes a bit advantage among these sort algorithms.

Then, in this experiment: We are using Binary Search for all remain cases.

That mean BS running time seems to be meaningless. And really it be.

So, look at this table: It is showing us time-performances of sort algorithms as well.

According to the result of last week homework, Quick Sort still take advantages and Radix Sort is runner-up as usual.