William Faunce

Carol Roberts

COS 225

13 April 2018

A8WilliamFaunce

Part I

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 35 | 15 | 100 | 90 | 20 | 88 | 50 | 25 | 55 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 35 | 50 | 90 | 20 | 88 | 10 | 25 | 55 | 100 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | 35 | 15 | 25 | 10 | 50 | 88 | 90 | 55 | 100 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 50 | 91 | 32 | 82 | 15 | 25 | 55 | 16 |

Part II

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 32 | 50 | 15 | 16 | 91 | 20 | 82 | 10 | 25 | 55 |
| 15 | 32 | 50 | 16 | 91 | 20 | 82 | 10 | 25 | 55 |
| 15 | 16 | 32 | 50 | 91 | 20 | 82 | 10 | 25 | 55 |
| 15 | 16 | 20 | 32 | 50 | 91 | 82 | 10 | 25 | 55 |
| 15 | 16 | 20 | 32 | 50 | 82 | 91 | 10 | 25 | 55 |
| 10 | 15 | 16 | 20 | 32 | 50 | 82 | 91 | 25 | 55 |
| 10 | 15 | 16 | 20 | 25 | 32 | 50 | 82 | 91 | 55 |
| 10 | 15 | 16 | 20 | 25 | 32 | 50 | 55 | 82 | 91 |

* 1. In my testing selection sort was slightly faster than insertion sort. This is because in the insertion sort method it uses a while loop (line 48 of SortMethods.java) that will insert the data value into its correct index where the previous element is less than and the next element is greater than the value.
  2. In my testing the runtime of insertion sort on sorted data (in the best case) is vastly faster than random data. This is because when the array is filled with random numbers in lines 76-81 the numbers getting inserted are completely random so it will take more time for insertion sort to run through the numbers comparing one to another.
  3. The runtimes of insertion sort on backwards data compared to random shows that the random data scenario is faster than the worst case scenario because when the array is filled in lines 49-54 it has to work backwards through the array instead of forwards. This increases the amount of time it takes to sort the whole array.

Screenshots:



