Insertion Sort:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Time | Average Time | Maximum Time |
| Sorting 1000 integers | 0 | 0.11 | 2 |
| Sorting 10000 integers | 9 | 9.78 | 12 |

[22, 90, 36, 72, 75, 94, 55, 7, 77, 45]

[7, 22, 36, 45, 55, 72, 75, 77, 90, 94]

Min time: 0

Average time: 0.11

Max time: 2

Min time: 9

Average time: 9.78

Max time: 12

d) it increase by 10-fold

Selection Sort:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Minimum Time | Average Time | Maximum Time |
| Sorting 1000 integers | 0 | 0.29 | 2 |
| Sorting 10000 integers | 23 | 24.15 | 53 |

22 65 43 42 73 68 22 68 15 36

15 22 22 36 42 43 65 68 68 73

Min time: 0

Average time: 0.29

Max time: 2

Min time: 23

Average time: 24.15

Max time: 53

d) it increase by 100-fold

3. Compare the average running times of insertion sort and selection sort. Which algorithm performs better? Can you give any reason?

Insertion sort is faster than selection sort. Insertion *sort* scans as many elements as it needs, while selection sort must scan all remaining elements to find the k+1st element.