**Speed Comparisons**

Speed Table for Random Addition

|  |  |  |
| --- | --- | --- |
| Data Size (number of entries) | Time Polymorphic Tree | Time Java TreeMap |
| 5,000 | 29 | 10 |
| 10,000 | 31 | 9 |
| 15,000 | 24 | 23 |
| 17,000 | 41 | 24 |
| 20,000 | 36 | 32 |

Speed Table for Sequential Addition

|  |  |  |
| --- | --- | --- |
| Data Size | Time Polymorphic Tree | Time Java TreeMap |
| 5,000 | 233 | 25 |
| 10,000 | 970 | 12 |
| 15,000 | 2472 | 8 |
| 17,000 | 3042 | 13 |
| 20,000 | 4869 | 21 |

**Explanation of Results**

The above tables highlight the differences in performance between Polymorphic Trees and Java’s TreeMap in addition to performance differences between using sequential vs. random addition. We see that the time associated with building a polymorphic tree grows much faster when sequential, as opposed to random, values are used. This may be partly because sequential addition ensures there will be no random values. Hence, the corresponding tree will be significantly larger. In the case of random number insertion however, the amount of time the tree building takes increases on roughly the same linear order for both polymorphic trees and Java’s TreeMap.