Your report goes in this file. Remove this description and replace it with your report. The report consists of two parts:

1. Two tables showing speed comparison between polymorphic tree and Javas’ TreeMap. Use TreeSpeed.java for information on how to obtain time information. Each table should have two columns: data size (number of values used) and the time (in milliseconds). Each table should have at least five entries. The first table will show results for trees created with numbers in a sequence and the second table with trees created with random numbers.

PolymorphicBST Tree

|  |  |
| --- | --- |
| 5000 Enteries (random numbers between 1 and 500000) | 46 (msec) |
| 5000 Enteries (in order from 1-5000) | 203 (msec) |
| 10000 Enteries (random numbers between 1 and 500000) | 48 (msec) |
| 10000 Enteries (in order from 1-10000) | 906 (msec) |
| 50000 Enteries (random numbers between 1 and 500000) | 148 (msec) |

Java TreeMap

|  |  |
| --- | --- |
| 5000 Enteries (random numbers between 1 and 500000) | 8 (msec) |
| 5000 Enteries (in order from 1-5000) | 6 (msec) |
| 10000 Enteries (random numbers between 1 and 500000) | 11 (msec) |
| 10000 Enteries (in order from 1-10000) | 9 (msec) |
| 50000 Enteries (random numbers between 1 and 500000) | 63 (msec) |

1. Two or three lines explaining the table results.

The PolymorphicBST tree is better at adding things to a tree in a random order. When we added 5000 entries of random numbers between 1-50,000 it took 46 msec whereas the 5000 entries in order from 1-5000 took 203 msec. Java’s TreeMap performed both the random and in-order test better than the PolymorhphicBST Tree but it added entries in order faster than adding entries in a random order.