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.

In Sequence

|  |  |
| --- | --- |
| Data Size | Time (milliseconds) |
| (PolyTree) 100 | 0 |
| (PolyTree) 5000 | 238 |
| (PolyTree) 20000 | 3331 |
| (TreeMap) 100 | 0 |
| (TreeMap) 5000 | 1 |
| (TreeMap) 20000 | 5 |

Random

|  |  |
| --- | --- |
| Data Size | Time (milliseconds) |
| (PolyTree) 100 | 6 |
| (PolyTree) 5000 | 32 |
| (PolyTree) 20000 | 22 |
| (TreeMap) 100 | 1 |
| (TreeMap) 5000 | 16 |
| (TreeMap) 20000 | 48 |

1. After running the tests and examining the data set, it shows that the TreeMap data structure runs significantly more efficiently than the Polymorphic Tree architecture when adding elements in order. I’m not entirely sure, but I think there is a difference in speed because the TreeMap utilizes Nodes in its implementation while the Polymorphic Trees essentially call themselves recursively. The speed in both TreeMaps and Polymorphic Trees are Big O(logn).