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.

|  |  |  |
| --- | --- | --- |
| Data Size | Avg. Time in Milliseconds (PolyTree / JavaTree) | Sequential Numbers |
| 2500 | 36/2 |
| 5000 | 147/4 |
| 10000 | 612/5 |
| 15000 | 1413/6 |
| 20000 | stackoverflow |

|  |  |  |
| --- | --- | --- |
| Data Size | Time in MS | Random Numbers |
| 2500 | 7/4 |
| 5000 | 11/6 |
| 10000 | 13/8 |
| 15000 | 18/12 |
| 20000 | 21/x |

1. Two or three lines explaining the table results.

As data input increases, the output results from TreeSpeed.java show a trend of increasing time.

The performance time for the polymorphic BST was comparatively worse, with the memory overflowing with an input of 20000. One reason for this could be because all the data was added sequentially from to the right tree, causing it to overflow.