|  |  |  |
| --- | --- | --- |
| List size | Time taken by | Time taken by |
|  |  |  |
| 25,000 | 790.6666666666666 | 190 |
| 50,000 | 2925 | 339.6666666666667 |
| 100,000 | 11283 | 675.3333333333334 |
| 200,000 | 44576.666666666664 | 1311 |
| 400,000 | 177913.33333333334 | 2603 |

as each fraction is getting closer to 3.99 so we round it up to 4

*Explanation:* The inner loop is executed times (the size of the list), and each time to find an element at index, get begins at the first node of the list until it finds the position. The statement element = list.get(i) ranges from 0 to in the previous for loop, so the total execution time for get is proportional to which is proportional to . Therefore, the order growth of is .

as each fraction is getting closer to 1.99 so we round it up to 2

*Explanation:* The while loop runs through iterations and during each iteration takes a constant amount of time as the statement Iterator’s hasNext() determines whether the list has more elements left to process instead of starting at the beginning of the list like get().

Therefore, the order growth of is

*Conclusion:* is more efficient because .