|  |  |  |
| --- | --- | --- |
| List Size | Time taken by get() | Time taken by next() |
| n | T(n) | T’(n) |
| 50,000 | 3,369 ms | 53 ms |
| 100,000 | 13,161 ms | 117 ms |
| 200,000 | 52,562 ms | 196 ms |
| 400,000 | 209,497 ms | 394 ms |

⋄ What, approximately, are the ratios T(2n)/T(n) (the middle column)?

Between 50,000 & 100,000: ~3.9x

Between 200,000 & 100,000: ~3.9x

Between 400,000 & 200,000: ~3.9x

⋄ What, approximately, are the ratios T′(2n)/T′(n) (the third column)?

Between 50,000 & 100,000: ~2.2x

Between 200,000 & 100,000: ~1.6x

Between 400,000 & 200,000: ~2.0x

⋄ Given the measurements obtained, what is your best guess about the order of  
growth of T(n)?

Each new doubling of n will yield 3.9 times the milliseconds.

⋄ Given the measurements obtained, what is your best guess about the order of  
growth of T′(n)?

Each doubling of n will yield between 1.6 times and 2.2 times (avg 1.9) times the milliseconds.

⋄ Which method is more efficient, list.get() or iterator.next()? Why?

Definitely iterator.next()! List.get() has to run through each element in linear time. Worst case scenario, if it needs to get an item all the way at the end of the list, it will run O(n). Iterator.next() is constant time or very close to it. It will save the last element, then the next time the method is called, just advance the pointer instead of starting from the beginning.