**Part 1 – Performance of Searching Routines**

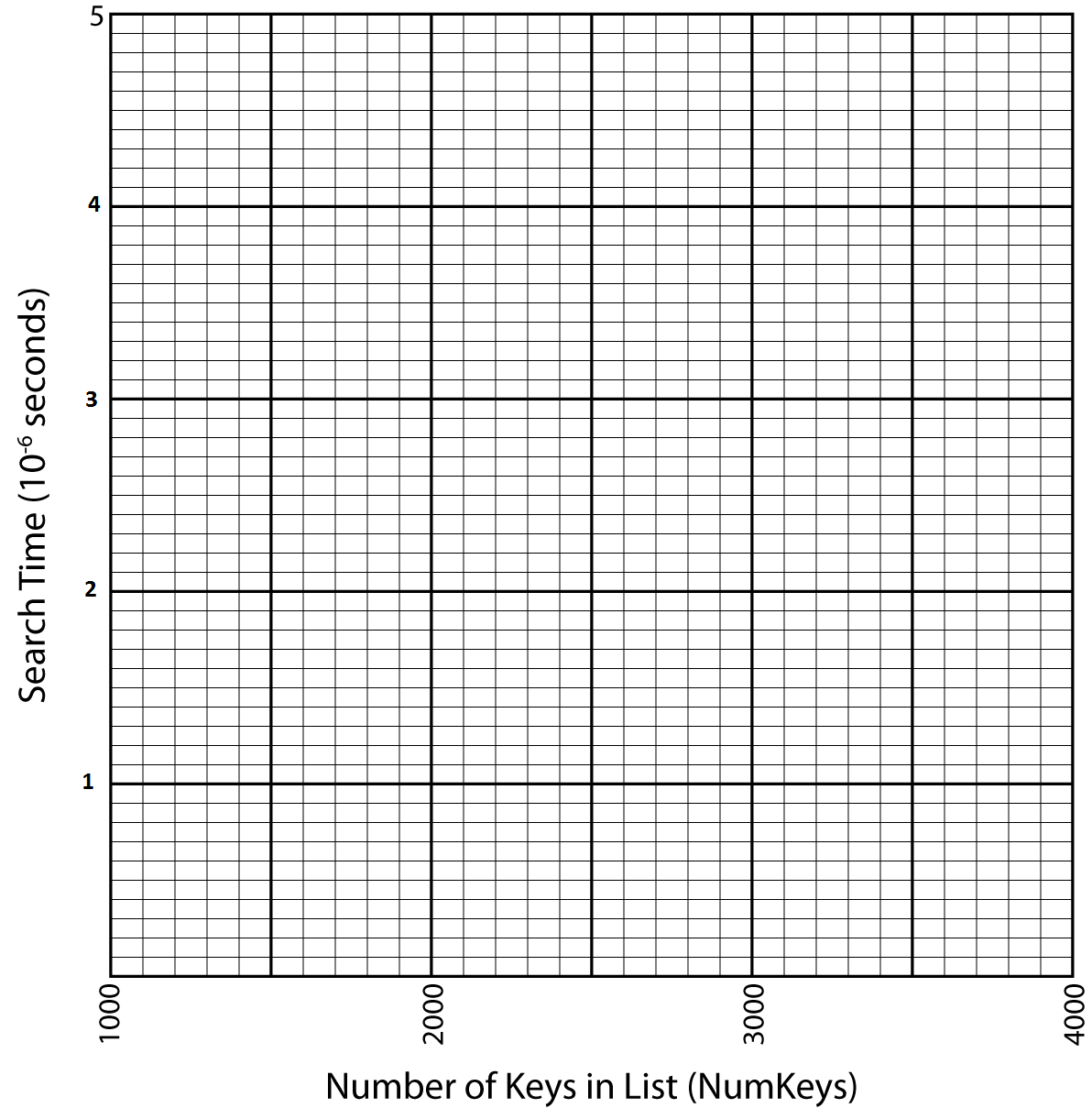
The following questions are taken from *A laboratory Course in C++ Data Structures*, by Roberge, Brandle and Whittington:

**Step 2**: Complete the following table by recording the **Time per Search** of the linearSearch(), binarySearch(), and unknownSearch() routines for each of the values of numKeys listed in the table.

**Execution Times of a Set of Searching Routines**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Routine | Number of Keys | | | |
| 1000 | 2000 | 3000 | 4000 |
| linearSearch() *O( N )* |  |  |  |  |
| binarySearch() *O( logN )* |  |  |  |  |
| unknownSearch() *O( )* |  |  |  |  |

**Step 3**: Plot the results below



**Step 4**: Consider the two routines in the table above with known **big O** execution times. Which one should run faster? Describe the expected shape of the curve for both algorithms.

**Step 5**: Using the code in the file search.h and your measured execution times as a basis, what is the execution time (Big O) of the unknownSearch() routine. Briefly explain your reasoning behind this estimate.

**Part 2 – Performance of Sorting Routines**

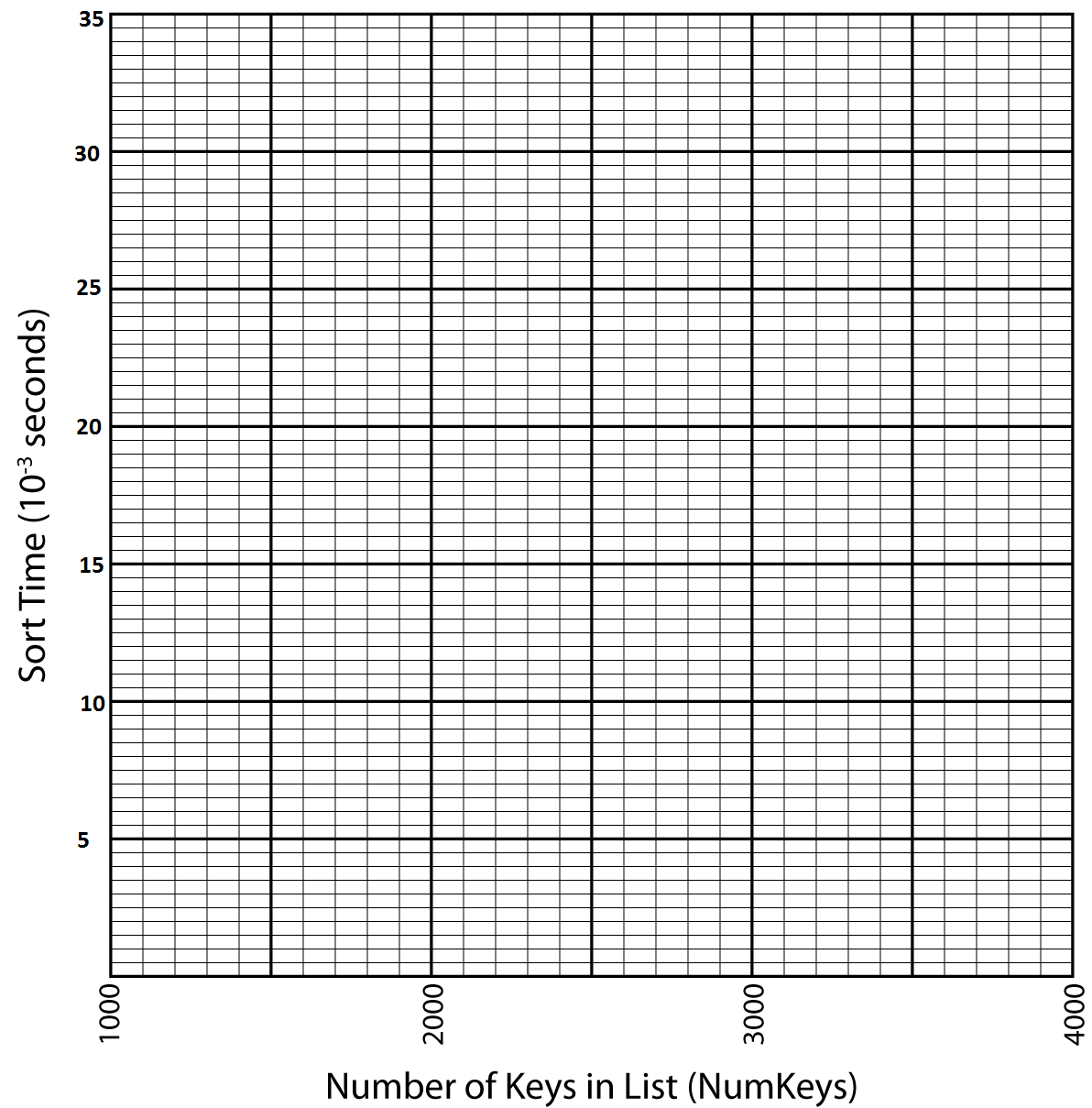
The following questions are taken from *A laboratory Course in C++ Data Structures*, by Roberge, Brandle and Whittington:

**Step 2**: Complete the following table by recording the **Time per Sort** of the selectionSort(), quickSort(), and unknownSort() routines for each of the values of numKeys listed in the table.

**Execution Times of a Set of Sorting Routines**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Routine | Number Of Keys | | | |
| 1000 | 2000 | 3000 | 4000 |
| selectionSort() *O( N2 )* |  |  |  |  |
| quickSort() *O( N logN )* |  |  |  |  |
| unknownSort() *O( )* |  |  |  |  |

**Step 3**: Plot the results below



**Step 4**: Consider the two routines in the table above with known **big O** execution times. Which one should run faster? Describe the expected shape of the curve for both algorithms.

**Step 5**: Using the code in the file sort.h and your measured execution times as a basis, what is the execution time(Big O) of the unknownSort() routine. Briefly explain your reasoning behind this estimate.