# Lab 10

## A study of sorting algorithms and their performance

Investigate the run-time and space complexities for the following sorting algorithms. When filling in the “Best Case,” “Average Case,” and “Worst Case,” do so by using Big-O notation, e.g., ***and*** by describing the type of data that leads to this case, e.g., “This occurs when the data is already sorted,” or “This occurs when the data is originally in reverse-sorted order.” The “Resource” column is for citing your resources, e.g., “Our textbook, page …” or <https://some.incredible.site.org>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Run-time (Time) Complexity | | | | |
| Algorithm | Best Case | Average Case | Worst Case | Resource |
| Bubble Sort |  |  |  |  |
| Selection Sort |  |  |  |  |
| Insertion Sort |  |  |  |  |
| Merge Sort |  |  |  |  |
| Quick Sort |  |  |  |  |
| Radix Sort |  |  |  |  |

With the space complexity table, just fill in your answers using Big-O notation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Space Complexity | | | | |
| Algorithm | Best Case | Average Case | Worst Case | Resource |
| Bubble Sort |  |  |  |  |
| Selection Sort |  |  |  |  |
| Insertion Sort |  |  |  |  |
| Merge Sort |  |  |  |  |
| Quick Sort |  |  |  |  |
| Radix Sort |  |  |  |  |