**Gerrald Andersen**

**Lab 07 CPSC 1050**

**October 27th, 2025**

1. ✓
2. ✓

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Example | Data Description | Selection Sort | Bubble Sort | Quick Sort |
| 1 | Completely Random | 28 | 27 | 26 |
| 2 | Completely Random | 120 | 75 | 66 |
| 3 | Ascending Sorted | 120 | 15 | 150 |
| 4 | Descending Sorted | 120 | 120 | 132 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Example | Data Description | Selection Sort | Bubble Sort | Quick Sort |
| 5 | Completely Random | 105 | 105 | 73 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Example | Data Description | Selection Sort | Bubble Sort | Quick Sort |
| 6 | Completely Random | 105 | 84 | 78 |

1. The quickest algorithm is Quick Sort and the slowest algorithm is Selection Sort.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Value | Sequential Search | | Binary Search | |
| Found? | # of tries | Found? | # of tries |
| 3 | Not Found | 16 | Not Fount | 4 |
| 16 | Found | 2 | Found | 3 |
| 99 | Found | 11 | Found | 4 |
| 300 | Found | 16 | Found | 5 |
| 301 | Not Found | 16 | Not Found | 5 |

1. The Binary Search algorithm uses the fewest number of comparisons overall around 3-5 tries.  
   But, Binary Search can only be used if the data is sorted.
2. ✓
3. ✓