**Junior Knights - Course 2 Week 1 Assignment: Sorting!!!**

**Solution**

|  |  |  |
| --- | --- | --- |
|  | Bubble Sort | Selection Sort |
| 10,000 | 1 s | 0 s |
| 20,000 | 3 s | 1 s |
| 50,000 | 14 s | 5 s |
| 80,000 | 37 s | 14 s |
| 100,000 | 58 s | 21 s |

1. Is there a general pattern between the input size of the array and how long the each of these sorts takes to run?

There is a pattern, but depending on your computer’s specs/processes it may be harder to tell. In a perfect world the pattern you should see is that based on an input size n, both bubble sort and selection sort take n2 units of time on average. Roughly speaking, if we double the input size, the run time gets multiplied by 4 (which is 2 squared). In general, if we multiply the input size by k, the corresponding runtime is multiplied by k2.

1. Which sort is faster?

Selection Sort – the reason for this is that selection sort only does one swap per iteration, while Bubble Sort potentially does many. The savings in time is avoiding those extra swaps. But, both algorithms shuffle through the same amount of data.

1. Is it significantly faster?

Yes, especially as input sizes grow larger. Look at the difference between runtime of the two at 10,000 compared to the difference at 100,000. Though the truth is that the answer to this question depends on how “significantly” is determined. If you take a careful look at the numbers, the ratio of the times is roughly constant on each row. Though the bubble sort takes more time, no matter what n is, it only takes about 2.6 to 2.7 times more time. For a larger value of n, such as 1,000,000, both of these sorts would take quite a long time.