Algorithm efficiency lab Madison Hubbard

Fibonacci:

|  |  |  |  |
| --- | --- | --- | --- |
| n-th term | value | Recursive | Iterative |
| 10 | 55 | 2 milliseconds | 5 milliseconds |
| 35 | 9227465 | 1,608 milliseconds | 5 milliseconds |
| 22 | 17711 | 9 milliseconds | 4 milliseconds |

The recursive one is slower than the iterative one because: it calls itself many times because it needs the index before it to calculate the one that it is at and so on and so on until it gets to the desired nth term, which makes the recursive one be slower.

Big-O efficiency for recursive Fibonacci: O(2^n)

Sorting:

|  |  |
| --- | --- |
| List size | Time to sort |
| 1000 | 532 milliseconds |
| 5,000 | 27,605 milliseconds |
| 10,000 | 153,484 milliseconds |
| 15,000 | 318,891 milliseconds |

The efficiency for this one seems like O(n) because it iterates through all the items.

Big-O efficiency for bubble sort: O(n)

Searching:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| List size | Find… | Your search 1 | Your search 2 | Binary Search |
| 999,999 | 977,777 | 837 milliseconds | 529 milliseconds | 3 milliseconds |