PFII Lab 10: Recursion II

In this lab I learned more about recursion application when calling functions, especially with relation to arrays. Another thing I learned in this lab is the change in run-time efficiency when implementing these functions using recursion compared to iteration. The results (as shown below) give us a representation of how much more efficient the recursive Maximum Subsequence Sum (Lab 10) is compared to the iterative Maximum Subsequence Sum (Lab 2). Similar is true for the different methods of searching where Binary search is much more efficient than linear search (O(n) vs O(nlogn).

**Maximum Subsequence Sum times (clock ticks)**

|  |  |  |
| --- | --- | --- |
| **n** | **Lab 2 MSS** | **Lab 10 MSS** |
| 1000 | 5153295 | 69739 |
| 10,000 | 516244521 | 1027910 |
| 100,000 | 5.42057E10 | 10609276 |
| 1,000,000 | 5.27421E12 | 126423276 |

**Linear vs. Binary Search when element is in the array times (clock ticks)**

|  |  |  |
| --- | --- | --- |
| **n** | **Linear Search** | **Binary Search** |
| 1000 | 2937 | 94 |
| 10,000 | 28596 | 129 |
| 100,000 | 197332 | 171 |
| 1,000,000 | 2032567 | 209 |

**Linear vs. Binary Search when element is NOT in the array times (clock ticks)**

|  |  |  |
| --- | --- | --- |
| **n** | **Linear Search** | **Binary Search** |
| 1000 | 3530 | 120 |
| 10,000 | 41211 | 138 |
| 100,000 | 483047 | 206 |
| 1,000,000 | 5165930 | 241 |

*\*code was run on onlineGDB*