1. Using Project 2, time each of the functions of Bubble Sort and Selection Sort and compare their times. Do the same for Linear Search and Binary Search functions. Provide screenshots showing the times (in seconds). Which algorithm would be best if there were 10,000 or 100,000 or 1 million values (consider for both sorting and searching)?

Text

Description automatically generated

When it comes to bubble sort and selection sort, selection sort seems to be faster when it comes to sorting through a certain number of values (based on running through a smaller test case). You can see that bubble sort took 722 microseconds where selection took 275. When it comes to the linear or binary search it would be faster to use a binary because it looks for a value based on which half of the total size the value resides. A linear search must run through each number compared to dividing the size into small chunks to fine the value. The picture shows that it took 626 microseconds to do a linear search compared to 190 seconds for a binary.

1. Fill out the following table about the different types of algorithms discussed this semester.

|  |  |  |
| --- | --- | --- |
| **Category** | **Description of How it Works** | **Examples** |
| Greedy Algorithms | Takes what it can when it can. Will always take the lowest possible path at that point in time even in the long run its not lowest. | Money denominations to break a value into perspective bills.  Huffman encoding gives more frequent characters smaller code and the less often bigger. |
| Divide and Conquer Algorithms | Takes a problem and divides it into subsequent problems. Can take those and divide them even further to small subproblems then combines the solutions in the end. | Binary sort  Quick sort  Merge sort  Selection problems  Matrix multiplication |
| Dynamic Programming Algorithms | Any program that can represented as a mathematical formula can be written recursively. Dynamic programming is a recursive algorithm written to be non-recursive so that its subproblems can be written recursive. | Fibonacci numbers  Coin Change that could have almost an infinite number of solutions. |
| Brute Force Algorithms | Algorithm that doesn’t process, only does checks then moves to the next letter. If the letter doesn’t match it +1 the letter, then continues until the letter is found. | Sequential search  Selection sort  Bubble sort |
| Randomized Algorithms | Any sort of algorithm that has some sort of randomness to it like random input, to save time. | Las Vegas algorithm  Monte Carlo algorithm  Randomized binary search  Random number generators  Amplification  CAPTCHA |
| Backtracking Algorithms | An algorithm that tries a solution but if it fails backtracks to try a different solution. | Arranging furniture in a house  Tic Tac Toe |