Searching Algorithms

By: Oliver Mo and Michael Karimizadeh

|  |  |
| --- | --- |
| Linear Search Advantages | Linear Search Disadvantages |
| * Simple and easy to implement * Data can be stored in any order * Memory efficient | * Inefficient because it requires to search through the whole array if needed. * Less efficient in large data sets. * Either runs longer even after it finds the index(for loop) or creates an infinite loop if it doesn't find the index(while loop) |

|  |  |
| --- | --- |
| Binary Search Advantages | Binary Search Disadvantages |
| * Fast when sorted * Efficient with large data sets * Has a worst case of O(log n) | * Difficult to implement and prone to error. * If not sorted, can cause errors. |

**Linear search pseudo code**

Create for loop

If first in array is not equal to integer searched for, repeat for next in array

If second in array is not equal to integer searched for, repeat for next in array

Repeat pattern until value is found

Display index at which value is stored

**Binary search pseudo code**

Find high and low values

Create while loop

If high is still greater than low, continue; if not end loop

Guess a random index

Check if number at guessed index is greater than number searched for

If this is true change high to the guessed index -1

If first check in not true check if number at guessed index is less than num searched

If this is true change low to guessed index +1

If neither check is true then index guessed is correct index and stop checking

Repeat this pattern until index is found

Display index at which value is stored

**Execution Time**

|  |  |  |  |
| --- | --- | --- | --- |
| Search for number when number at: | Linear Search | Search for number when number at \_ out of a 50 array | Binary Search |
| 1 (Best Case) | 364 | 24, (Best Case) | 3277 |
| 5 | 728 | 12 | 3640 |
| 10 | 1092 | 6 | 4004 |
| 20 | 1198 | 0 (Worst Case) | 4369 |
| 30 | 1456 | 37 | 3640 |
| 40 | 1821 | 43 | 4004 |
| 50 (Worst Case) | 2184 | 49 (Worst Case) | 4369 |