INTRODUCTION TO ALGORITHMS

By Ligeia Bodden, Xinge Ji, and Sara Manning

What is an algorithm?

Processes in our lives that fit at least one criterion for being an algorithm:

Recipes.

The order of operations in math (PEMDAS).

Procedures for science experiments.

A precise set of instructions that solve a problem.

SEARCHING ALGORITHMS: the dictionary.

- Try to find the word “word” in the dictionary.
- Could go through each page, starting with page 1. If the word “word” is not on the page, go to the next page.

THIS IS CALLED LINEAR SEARCH AND IT TAKES A **VERY LONG TIME.**

- Could look at every other page.

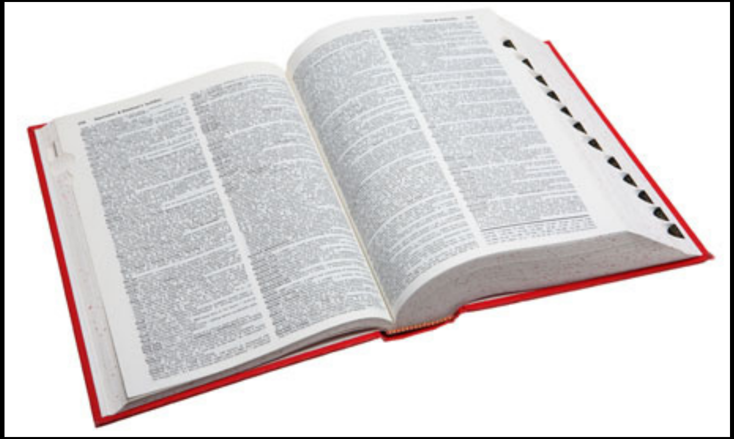
THAT IS STILL VERY TIME CONSUMING.

Binary search: the dictionary

1. Look in the middle of the dictionary.
    2. **If** the word "word" comes before the words on this page in the alphabet: second half does not contain "word."
        - **Else if** "word" comes after the words on this page in the alphabet: first half does not contain "word."
        - **Else:** the word "word" is on the page!

3. Look in the middle page of the half that the word "word" is in.

4. **While** the word "word" has not been found yet, repeat steps two through four.



https://www.flickr.com/photos/26878261@N07/7983928

BINARY SEARCH VS. LINEAR SEARCH (CONTINUED.)

For a 1,000 page dictionary...

In the worst case scenario, binary search would only require **ten** steps...
but linear search would take **1,000.**

If we doubled the number of pages...

In the worst case scenario, binary search would only require **eleven** steps...
but linear search would require **2,000.**

For 1,000,000,000 pages

In the worst case scenario, binary search would only require **31,622** steps...
but linear search would require **1,000,000,000.**

Binary search is a very efficient way to find a value in a list. It is very useful for programming and other areas of computer science.

HOWEVER, WE NEED TO SORT THAT INFORMATION FIRST...

Insertion Sort Algorithm

Taking items from an unsorted deck and inserting it into the sorted deck

Example deck:

{4 | 3, 2, 1}
{3, 4 | 2, 1}
{2, 3, 4 | 1}
{1, 2, 3, 4 }

Big-O of average insertion sort: $O(n^2)$, which means that if you have an array of 100 numbers, the computer runs 10 thousand times before it can present you with a fully sorted array

Merge Sort Algorithm

This is a recursive algorithm, meaning that this algorithm calls itself with simpler input values

Chopping the array in half recursively until each small array only contains 1 number, and then putting them back in order

Example array: {5, 3, 8, 7, 4, 0}
{5, 3, 8}{7, 4, 0}
{5, 3}{8}{7, 4}{0}
{5}{3}{7}{4}
{3, 5}

The Big-O of Merge Sort is O(n*log2(n)), meaning if you have an array of 9 elements, the computer runs 27 times before presenting you with the sorted array

Selection sort vs Merge sort

Differences:

They are two different ways to solve the same problem

Selection sort is a linear sorting algorithm while Merge sort is recursive, therefore Merge sort is more efficient than selection sort.

Selection sort takes up less hardware memory than Merge sort because there is many iterations in Selection sort while in Merge Sort there is more actual code.

Similarity:
Both are designed to sort an unsorted array

Bubble Sort

In bubble sort, the computer compares two adjacent items and swaps the items to place them in order. The computer follows this through all the way until all of the items are sorted.

For example: The numbers 9, 6, 3, 4 will be sorted in ascending order

The new arrangement is: 6, 9, 3, 4

The new arrangement is: 6, 3, 9, 4

The new arrangement is: 6, 3, 4, 9 -> The computer will then go back to the beginning

The new arrangement is: 3, 6, 4, 9

The final arrangement is: 3, 4, 6, 9

Selection Sort

In selection sort, the computer goes in ascending order to see if there are smaller items in an array. The computer then swaps the smallest value with the value it is checking. It then puts the smallest value behind a “wall” with the sorted items.

For example: The numbers: 9, 7, 3, 2 will be placed in ascending order

The new arrangement: 2/ 7, 3, 9

The new arrangement: 2, 3/ 7, 9 -> The computer will then go back and check to make sure the numbers are sorted

The final arrangement: 2, 3, 7, 9

What we thought about learning algorithms

Fun

fun

fun

FUN!

FUN

Live and Learn

There are many more algorithms other than the ones we have presented, such as Dijkstra’s algorithm, encrypting algorithms, graph traversing algorithms and etc. These are all topics that deserve further explanation and if you are interested feel free to explore on your own at home

Many thanks to our coordinators, Ms. Brossman, our guest speakers and our families!!

Thank you all for coming!