1. BigO Notation

tells the number of Operations (hence big O)

tells how fast the algorithm grows --> Rate of Growth

Eg - Binary Search needs logn operations to search n items, 2^x=n, x= logn to the base2

- running time --> Ologn

tells worst case run time

Eg - Binary search can never be slower than Ologn

Five major run times:

O(logn), O(n), O(n\*logn), O(n\*2), O(n!)

in the order of run times

O(n) time means you touch every element in a list once.

Examples:

O(logn) --> Binary Search

O(n) --> For loop

O(n\*logn)-->

O(n\*2) --> Nested for loops

o(n!) -->

2. Binary Search - only works for sorted items

Number of steps needed for binary search of n items:

2^x = n --> "x" steps i.e., log2 n steps

With each step the number of items to be searched for

gets reduced by half

3. Python list is a dynamic array

An array which is resizable.

##########Sorting algorithms##############

Sorting is important as a lot of algorithms work on sorted data

##########################################

4. Selection sort

Selection sort is an algorithm that selects the smallest element

from an unsorted list in each iteration and places that element

at the beginning of the unsorted list.

Sort by selection

5. Divide and Conquer

Base case

with each recursion - you reduce the sample size

6. Brute force vs Sliding window

Bruteforce Uses two for loops, sliding window slides across the input

O(n\*2) vs O(n)

7. An in-place algorithm is an algorithm which transforms input using no

auxiliary data structure.

8. Two Pointers Approach

Set one pointer in the beginning and one at the end and then

move them until they both meet.

Text, letter

Description automatically generated

To find the middle of linked list, use two pointers, one would move one step at a time, the other would move two steps at a time. By the time, the second pointer reaches end, the first pointer would reach the middle.

The pointers can start from anywhere and move at any pace and any direction.