Big O notation

algoritms is about finding the most efficient time save in running time.

Big O notation is a special notation that tells you how fast an algorithm is.

The run time in Big O notation is O(n).

Big O notation lets you compare the number of operations and how fast the algorithm grows.

Example

a linear search with 16 would take 16 operations O(16).

a binary search with 16 numbers would take O(log 16) = 4 operations.

Common Big O run times

O(log n), also known as log time. Binary search.

O(n), linear time. Simple search.

O(n · log n). A fast sorting algorithm, like quicksort.

$O(n^2)$. A slow sorting algorithm, like selection sort.

O(n!) A really slow algorithm, travelling person.

Recap

- Binary search is a lot faster than simple search
- O(log n) is faster than O(n), but it gets a lot faster once the list of items you're searching grows.
- Algorithms speed isn't measured in seconds.
- Algorithms times are measured in terms of growth of an algorithm.
- Algorithm times are written in Big O notation.