More BIG-O practice

Note that we have seen how to determine the big-o runtime of code, what operations on our data structures might we want analyze

* Insert (put things into the structure)
* Delete (take things out)
* Retrieve 1. By position: indexing

2. by name: searching

Other operations include: merge/union, min/max

Your first data structure: an array

Ann array is a sequence of objects (all of the same type) stored in consecutive memory locations.

By convention if we have n objects in the array they are stored in slot 0 through n-1

To add an object to the array, we would place it in slot n.

To search an array a for an element x we need, to use sequential search (a.k.a linear search)

Suppose we want to use an array to implement a set of objects

A set is an unordered collection of distinct (unique) elements.

To search the set, we are searching the array, which requires O(n) steps.

Check if x is a member: O(n)

Add x to the set: O(n)

Remove x from the set: O(n)🡪Take the last one and replace the target slot

Min/max: O(n)🡪need to search for everything