My Code is Slow… and I don't know why

An Introduction to Data Structures and Algorithms

CS 196 – 25 DotStar [Lecture M1]

NOTE

Don't worry if you don't understand all of this. This is the first part of a 3 microLesson series. We will be revisiting a lot of these concepts. You're also going to go over this stuff Ad Nauseam in CS 225.

Please stop us at any time to ask us questions.

What is a Data Structure

Simply, it's a structure that holds data.

You're familiar with:

- Variable
- Array
- Linked lists (coming soon to 125)

Today, we're going to focus on **Lists** and **Dictionaries**.

What is an Algorithm

English definition: a process or set of rules to be followed in calculations or other problem-solving operations

In computer science, and algorithm is a finite set of instructions, executed in order to achieve some end goal.

For example, a sorting algorithm is a finite set of instructions to sort a dataset.

What makes a good algorithm?

It **must** be correct.

It's better if it's **efficient**.

It's better if it **scales well**.

Big O Notation

Measure of how well an algorithm **scales**. Often confused with the **speed** of an algorithm. Although this isn't altogether inaccurate.

We usually use this to express how an algorithm's running time scales in relation to the input size (**n**).

Example notation:

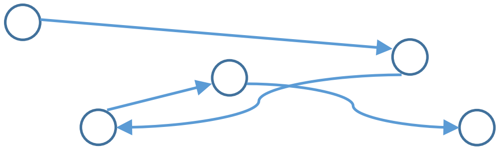
O(1) Time taken is independent of input size.
O(n) Time taken scales linearly with input size.
O(n^2) Scales quadratically with input size.

Think of a *for* loop over an array. The amount of time it would take to complete scales linearly with the size of the array.

Lists

The List Data Structure is exactly what you would expect it to be. Think of a list of things to buy, a list of things to do.

But let's get a little more creative. Could you see a treasure map being a list?



We'll talk more about how lists work next week. #GetHyped.

Dictionaries

Dictionaries are also close to what you would expect them to be. Think of an actual dictionary: You have a word which you look up, and you find a definition.

In CS, every dictionary has a **key** and a **value**. Think of calling a function with an argument, which then returns some information.

With a dictionary, you pass in a **key** and it returns a **value**.

What else can you think of that has a similar behavior?

Find this handout online at www.akmodi.com/docs/intro_to_alogos.pdf
Feel free to contact the lecturers: Abhishek Modi [akmodi2@illinois.edu] and Mark Miller [mrmillr3@illinois.edu]