Recursive Data Types

AN EXPLANATION FOR Introductory CS Students

based on CS 2102 - Matthew Keitelman

Data Types

what is a data type? a way to: store a value
work with it
programmatically as opposed to theoretically

how can we think of them?
essentially, they are like objects in our world
they can be stored and manipulated

Problem

how can we relate objects of data?
create a data type where the next data is
inside the first
(imagine russian dolls)
(but really, the first points to where the next is)

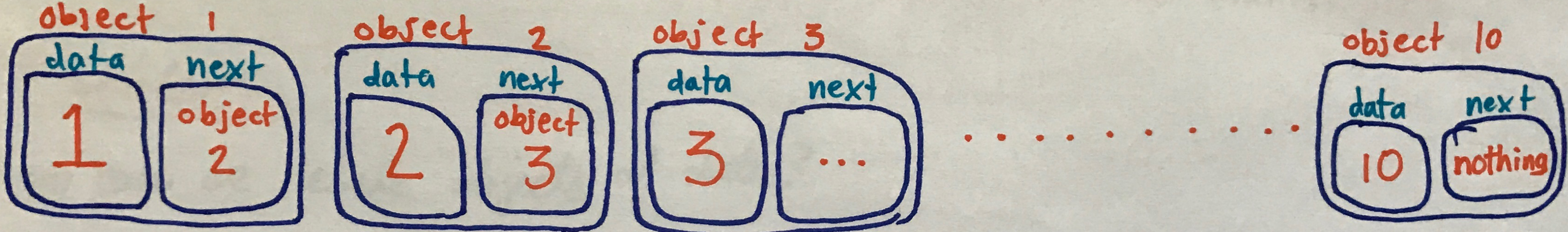
defining one in terms of itself or others is
RECURSION

thus, a recursive data type is one that has an attribute (or contains)
another data object of the same type

PROCESS

- begin at first data object
- follow the path of next objects
- keep going until no more remain

example: counting



- begin at first object: 1, 2, 3 . . .
- Follow the path:

we travel 1 → 2, 2 → 3, 3 → 4, . . .

how is this recursion?

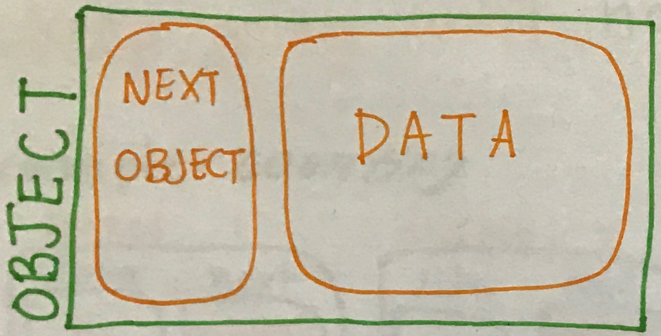
each data object is defined in terms of another.

object 1 is defined as data(1) and object 2

the last object must not be recursively defined (object 10) so that the chain can end.

Implementation

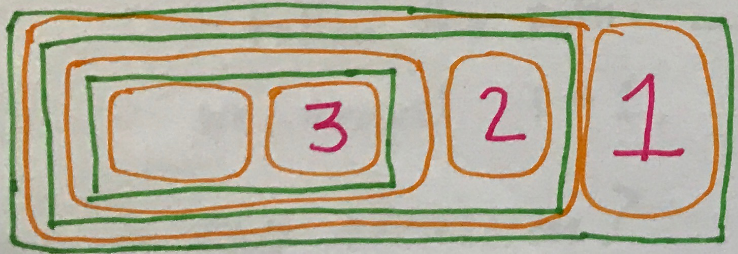
in programming languages, e.g. Python, an object is a concept where several attributes can be connected to one thing



DATA - the payload. from the counting example, a number such as 1 or 2 or 43

NEXT OBJECT - an entirely separate object. if DATA is 1, NEXT OBJECT's DATA should be 2, and so on.

Again, this is RECURSION. An object is defined in terms of another object.



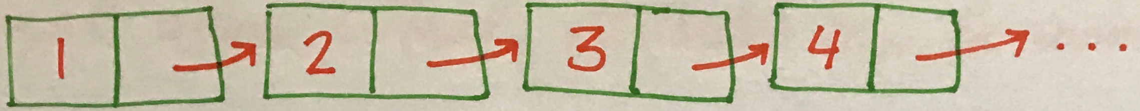
A recursive chain counting up from 1.

The empty yellow box has another object inside.

WARNING: FOR LARGE OR INFINITE (like all numbers) lists, we may run into stack overflow. The computer runs out of memory to keep track of all the levels deep it has gone and gives up!

USES

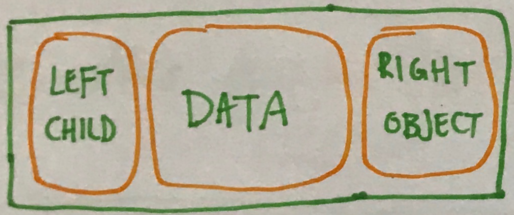
lists - an ordered collection of objects or values (eg. the numbers)

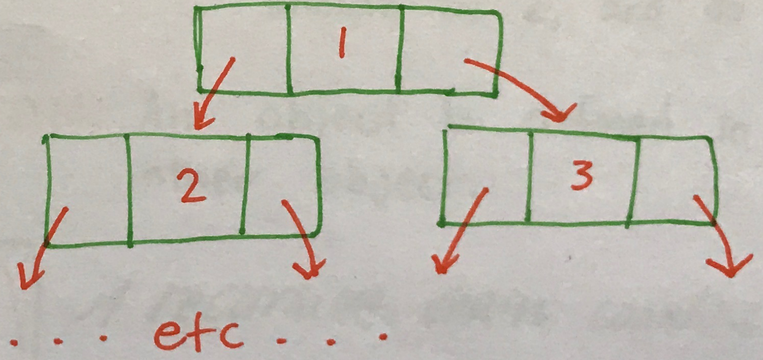


↑ another way of representing objects-within-objects

trees - a hierarchal organization, like a family tree

- requires 2+ "next objects" inside an object





eventually, the end of the list is achieved with a NULL POINTER - instead of linking to another pair, we're saying "this leads nowhere. stop here."