CPS 109 - Lab 3

Agenda

1 - Exhaustive Enumeration

2 - Indexing Lists

3 - Iterating

4 - Binary Numbers

5 - Bisection Search

A Handy Definition

Exhaustive Enumeration:

- Also known as “brute force” (most people call it this)
- Simply means you search through/compute *all* possibilities/items in a list or set.

Indexing Lists

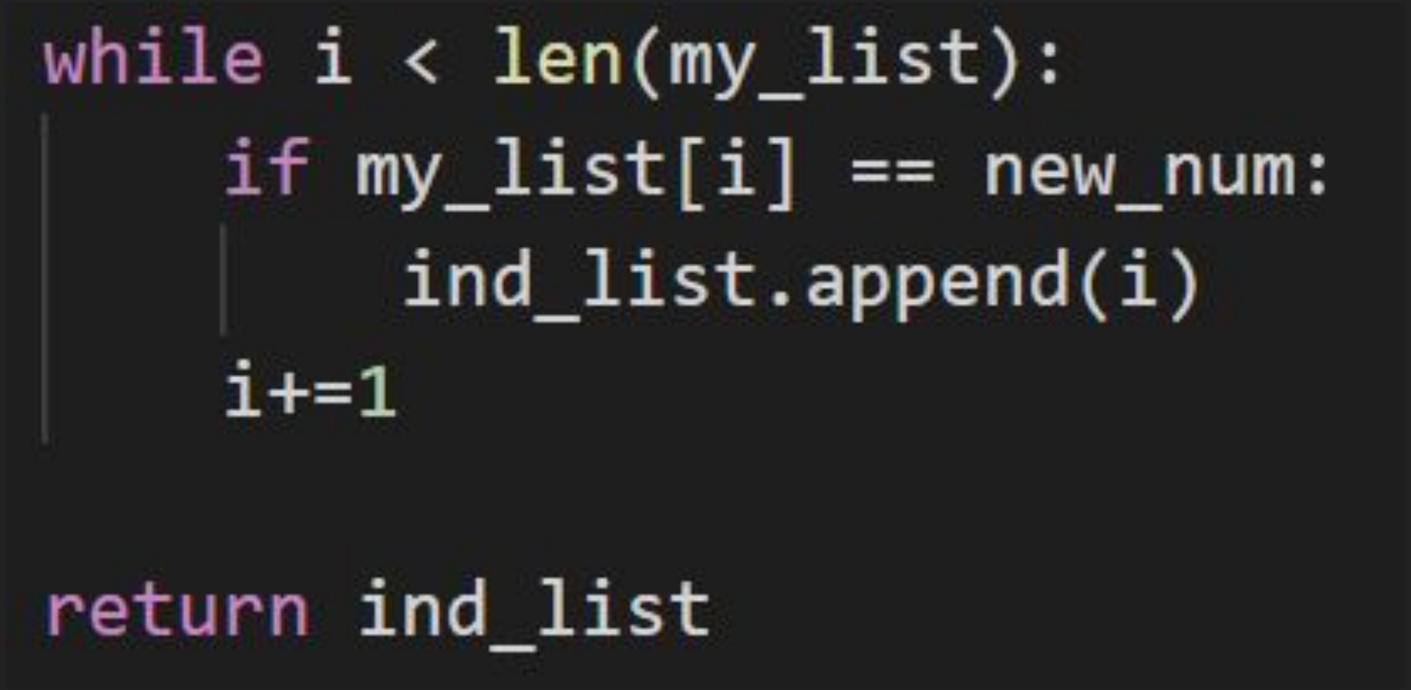
Remember how we went over loops last week? And we just talked about exhaustive enumeration? It’s handy in the context of indexing lists.

What’s an index? An index is the integer location of an item in a list or string.

Iterating

To iterate through something (usually a list or a string), you are simply accessing it by jumping from one index to the next.

Example of Iterating/Indexing



```
while i < len(my_list):
    if my_list[i] == new_num:
        ind_list.append(i)
    i+=1

return ind_list
```

Bisection Search

In this lab, it’s just a way to find roots for a given number (for now).

It closes in (very slowly) on what the approximate root is for a given number. Think of it like a pendulum!

Binary Numbers

Author’s Note: No one works with binary numbers in Python.

Binary numbers are representations of real numbers in base 2.

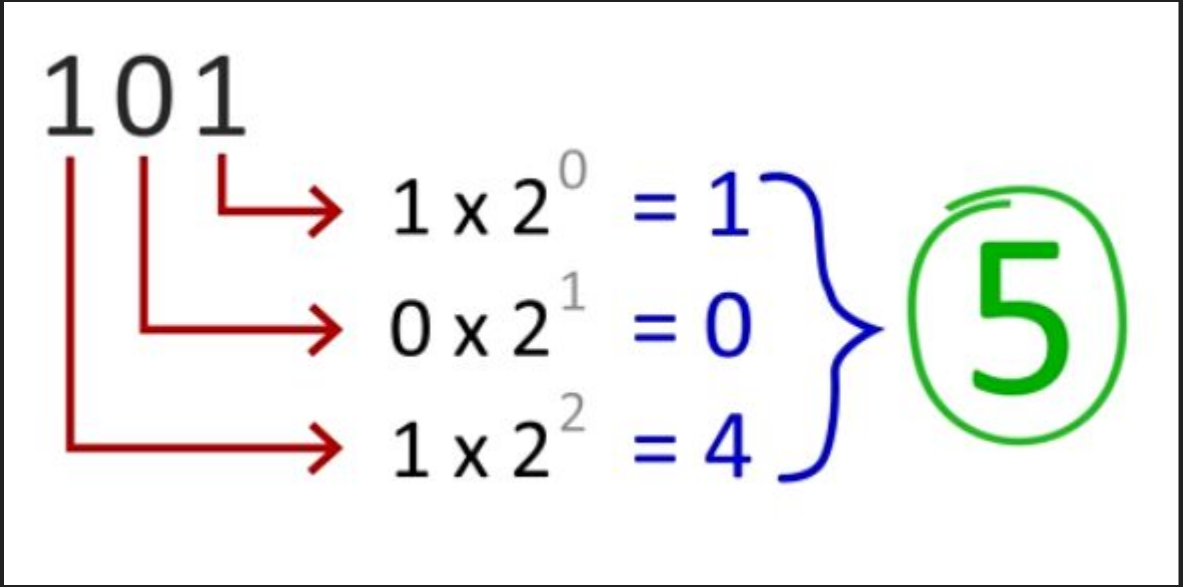
Side note: “Decimal” now refers to numbers in base 10.

Binary Numbers

You know those fancy numbers when they show computers on TV? 10101010… Those are binary numbers.

Converting between decimal and binary is pretty easy!

Binary Numbers



101

$1 \times 2^{0} = 1$

$0 \times 2^{1} = 0$

$1 \times 2^{2} = 4$

5

Binary Numbers

That last slide shows us how the 1s and 0s correspond with powers of 2. So, here is, essentially, how to count:

1 = 1

10 = 2

11 = 3

100 = 4

101 = 5

110 = 6