CPS 109 - Lab 4

“Bin” there done that (A quick review of Binary Search)

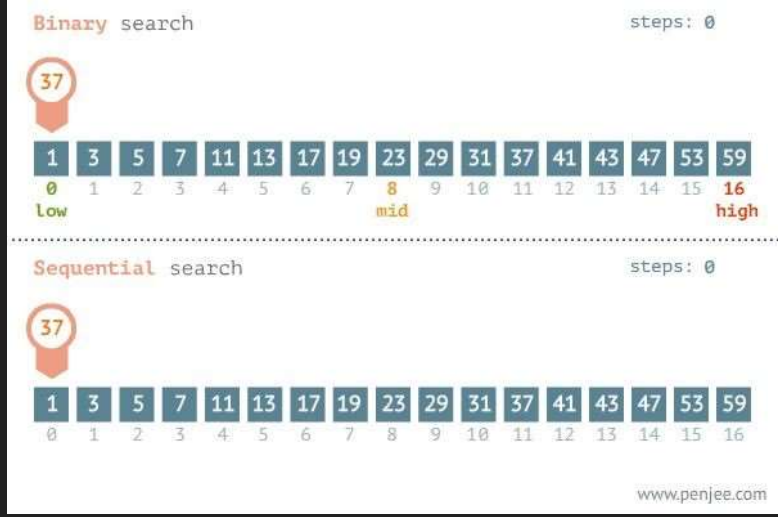
Before we set you off to work on your lab, we’ll give you a quick review of Binary Search.

Different kinds of searches and sorts are very important in computer science, as is the analysis of their runtimes

High Level Steps

1. Set mid to middle of a **sorted** array
   a. If the number at the middle is greater than the number we're looking for then restrict our search to the lower half
   b. If the number at the middle is less than the number we're looking for then restrict our search to the upper half
   c. If the number at the middle is the number we're looking for then return its index (done)
1. Continue 1. until we encounter c. or we have no range left (in which case, the number is not in the list).

To note: By halving our search space at each step, we are much more efficient than in a "naive" sequential search (but this requires a sorted array as input!)



Pseudocode Implementation

```
#Input: The list to search, the element to look for
#Output: The index of the element in the list, OR -1 if it is not in the list
def binary_pseudocode(input_list, element_to_look_for):
    low = 0, mid = 0, high = len(input_list)-1
    while an unsearched range still exists: # What does this mean?
        mid = the floor average of low and high # Recall syntax: //2
        if the item at mid == the element to look for:
            return the index of the item at mid
        if the item at mid > the element to look for:
            update the range to be the lower half of the list (update high)
        if the item at mid < the element to look for:
            update the range to be the upper half of the list (update low)
    return -1 # I.e. the element is not in the list
    # Super cool hint: What are low and high equal to at the end? Are they
    useful?
```

Binary Numbers

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

Binary numbers are representations of real numbers in base 2.

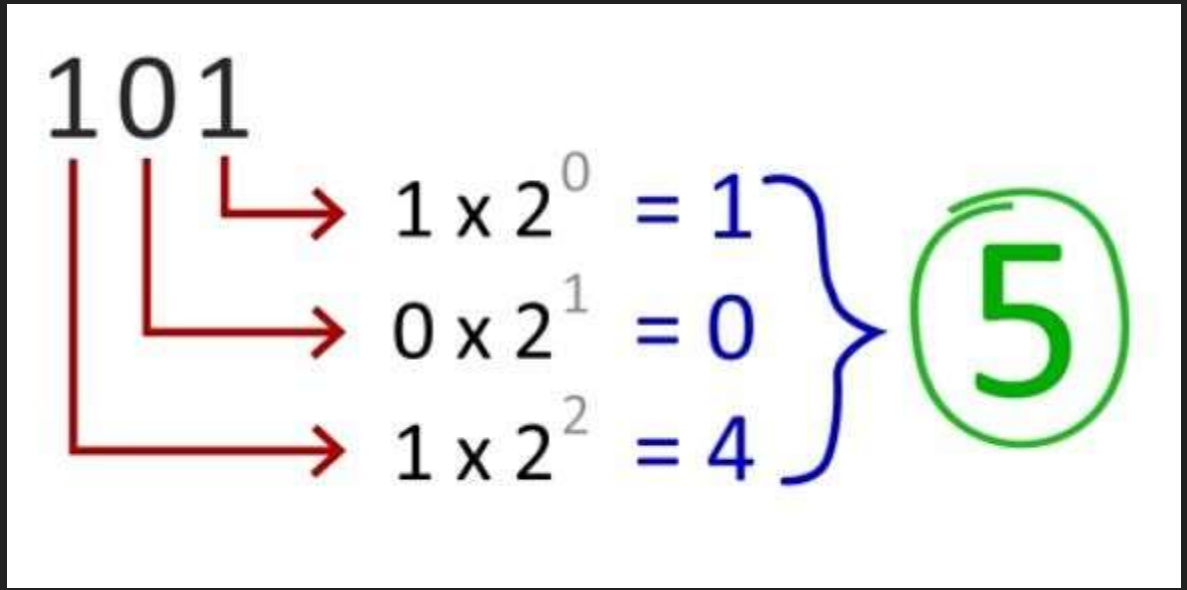
Side note: "Decimal" now refers to numbers in base 10.

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



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

Binary Numbers

Practice

Why don't we try some examples together?

Binary: 101
Decimal: 5

Binary: 111001
Decimal: 57

2**0 + 2**3 + 2**4 + 2**5

=

1 + 8 + 16 + 32 = 57

Decimal: 69
Binary: 1000101

What's the largest exponent of 2 that goes into 69?: 2**6
= 64 (Total: 64, Rem: 5)

What's the next largest that goes into 5?: 2**2
= 4 (Total: 64 + 4 = 68, Rem: 1)

What's the next largest that goes into 1?: 2**0
= 1 (Total: 68 + 1) = 69

Lab #4 Quiz Task)

Print out the first N integers, one per line, where N is read in from the user. When printing the numbers, apply the following rules:

1. For multiples of 3, print "Fizz" instead of the number.
2. For multiples of 5, print "Buzz" instead of the number.
3. For numbers that are multiples of both 3 and 5, print "FizzBuzz".

For example, if the user enters N=15, the output should be:

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
```

- If the user enters a number less than 1, print out “N must be greater than 1”
- If the user enters a number greater than 100, print out “Too much work, no thanks”