BEFORE WE START . . .

PYTHON/JUPYTER SETUP

ASSIGNMENT 1

QUIZ 1

GEOTAB
FLOATING POINT NUMBERS
WHY SHOULD YOU CARE?

1. Fundamental to all calculations on a computer
2. It works differently than pen and paper
3. Important to know the limitations
4. Calculations can be slightly or completely wrong
<table>
<thead>
<tr>
<th>BINARY SYSTEM</th>
<th>DECIMAL SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>100100</td>
<td>36</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>-101</td>
<td>-5</td>
</tr>
<tr>
<td>0.01</td>
<td>0.25</td>
</tr>
<tr>
<td>0.010101010...</td>
<td>0.333333333...</td>
</tr>
<tr>
<td>0.000110011...</td>
<td>0.1</td>
</tr>
</tbody>
</table>
BITS AND BYTES

Smallest amount of information a computer can store is 1 bit, a 0 or 1.

8 bits are one byte

2 bytes are one word
5.1e+309 = 5.1 \cdot 10^{309}
IEEE754 STANDARD FOR FLOATING POINT NUMBERS

64 bits

52 bits for the mantissa
11 bits for the exponent
1 bit for the sign
WE CAN WRITE ANY REAL NUMBER WITH A SIGN, EXPONENT+MANTISSA

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]
WE CAN WRITE ANY REAL NUMBER WITH A SIGN, EXPONENT+MANTISSA

\[ 2^{e-1023} \cdot (1 + m) \]
EXAMPLES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

\[ s = 0 \]
\[ e = 1023 \]
\[ m = 0.5 \]

\[ x = 1.5 \]
EXAMPLES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

\[
\begin{align*}
s &= 0 \\
e &= 1020 \\
m &= 0.5
\end{align*}
\]

\[ x = 0.1875 \]
EXAMPLES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

- \( s = 0 \)
- \( e = 1024 \)
- \( m = 0 \)

\( x = 2 \)
EXAMPLES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

\[ s = 0 \]
\[ e = 2040 \]
\[ m = 0 \]

\[ x \approx 1.4044 \cdot 10^{306} \]
EXAMPLES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

\[ s = 0 \]
\[ e = 2 \]
\[ m = 0 \]
**IMPORTANT SCALES**

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

**EXponent**

\[ 2^{11} = 2048 \]
\[ 2^{1024} \approx 10^{308} \]

Range: \[ 10^{-308} - 10^{308} \]
IMPORTANT SCALES

\[ x = (-1)^s \cdot 2^{e-1023} \cdot (1 + m) \]

MANTISSA / FRACTION

\[ 2^{-52} \approx 10^{-16} \]

Precision: 1 part in \(10^{16}\)
PLAY WITH FLOATING POINT NUMBERS YOURSELF

1) Jupyter notebook in PSCB57 repo

2) www.h-schmidt.net/FloatConverter/IEEE754.html
EXAMPLES!
ALGORITHMIC COMPLEXITY
def g2(x):
    if x==0:
        return 0
    if x==1:
        return 1
    return g2(x-1)+g2(x-2)

O(2^N)
## Algorithmic Complexity

<table>
<thead>
<tr>
<th>Time Complexity</th>
<th>Complexity Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(1)$</td>
<td>Constant</td>
</tr>
<tr>
<td>$O(\log(N))$</td>
<td>Logarithmic</td>
</tr>
<tr>
<td>$O(N)$</td>
<td>Linear</td>
</tr>
<tr>
<td>$O(N \log(N))$</td>
<td>Log Linear</td>
</tr>
<tr>
<td>$O(N^2)$</td>
<td>Quadratic</td>
</tr>
<tr>
<td>$O(N^3)$</td>
<td>Cubic</td>
</tr>
<tr>
<td>$O(2^N)$</td>
<td>Exponential</td>
</tr>
</tbody>
</table>
MATH
def fib(n):
    if n==0:
        return 0
    if n==1:
        return 1
    if n%2==0:
        fn = fib(n/2)
        fn1 = fib(n/2-1)
        return (2*fn1+fn)*fn
    if n%2==1:
        fn = fib((n+1)/2)
        fn1 = fib((n+1)/2-1)
        return fn*fn+fn1*fn1

\[ O(\log(N)) \]