Algorithm 2: Look through String Once

For a longer string, maybe we just want to look through it once?

initialize 27-bin histogram to all 0s
for each character in the string
**   increment the appropriate histogram bin**

But figuring out which bin to increment may be complicated.

Algorithm 3: Build a Bigger Histogram

What if we build a bigger histogram first:

initialize 128-bin histogram to all 0s
for each character in the string
**   increment bin for that character**
for each letter
**   add the two corresponding bins**
sum all non-letter bins

Now finding the bin is easy, but we need extra memory and computation.

Which Algorithm is Best?

Which approach is better?

What is the metric?
- Number of instructions executed?
- Number of clock cycles (time) required?
- Amount of memory needed?

Does our answer depend on the length of the string?

What if the string is sorted alphabetically?

Let's Pick Algorithm 2

The answer depends on the context and the application of our program.

We're going to go with Algorithm 2:

initialize 27-bin histogram to all 0s
for each character in the string
**   increment the appropriate histogram bin**

Why? Implementing the complex decision in the middle will be interesting.

