Lecture 14 — Problem Solving and Design, Part 1

Overview

This is the first of our lectures dedicated primarily to problem solving and design rather than on particular programming constructs and techniques

- Design:

    - Choice of container/data structure; choice of algorithm.

        - At the moment, we don't know too many containers, but we will think about different ways to use the one container - lists - we do know about.
    - Implementation
    - Testing
    - Debugging
- We will discuss these in the context of several variations on one problem:

    - Finding the mode in a sequence of values — the value (or values) occuring most often.
- There is no direct connect to a chapter in the text.
- We will start with a completely blank slate so that the whole process unfolds from scratch. This includes looking for other code to adapt.
- Working through problems like this is a good way to review what we've learned thus far.

Problem: Finding the Mode

- Given a series of values, find the one that occurs most often.
- Variation 1: is there a limited, indexable range of values?

    - Examples that are consistent with this variation include test scores or letters of the alphabet
    - Examples not consistent include counting words and counting amino acids
- Variation 2: do we want just the modes or do we want to know how many times each value occurs?
- Variation 3: do we want a histogram where values are grouped?

    - Example: ocean temperature measurements, pixel intensities, income values.

- In each of these cases, a specific value, the number of occurrences of a specific temperature, such as 2.314C, is not really of interest. More important is the number of temperature values in certain ranges.

Our Focus: A Sequence of Numbers

- Integers, such as test scores
- Floats, such as temperature measurements

Sequence of Discussion

- Brainstorm ideas for the basic approach. We'll come with at least two.

  - We will discuss an additional approach when we learn about dictionaries.
- Algorithm / implementation
- Testing

  - Generate test cases
  - Which test cases we generate will depend on the choice of algorithm. We will combine them.
- Debugging:

  - If we find a failed test case, we will need to find the error and fix it.
  - Use a combination of carefully reading the code, working with a debugger, and generating print statements.
- Evaluation

  - We can analyze using theoretical tools we will learn about later or through experimental timing

Discussion of Variations

- Frequency of occurrence:

  - What are the ten most frequently occurring values? What are the top ten percent most frequent values?
  - Output the occurrences for each value.
- Clusters / histograms:

  - Test scores in each range of 10
- Quantiles: bottom 25% of scores, median, top 25%