Stephen Belden

Chris Ruiz

Meghan Haukaas

**Problem 3 - Bin Packing: First Fit and Best Fit**

**Problem**

This problem asks us to place elements of a given size into bins of a given size, using two different algorithms. Bins and elements are both one-dimensional objects, represented by a single floating point number (their size).

Both of our algorithms must have a time class of θ(N log N), where N is the number of elements to be inserted. Each algorithm solves a different problem; first-fit inserts elements one-by-one as soon as they are known, and best-fit inserts elements after the entire set of elements to be inserted is known.

We need to design a container that represents a bin. We also need a way of creating new bins of a defined or default size on demand. To accurately test the time complexity of these algorithms, we will need a way to generate random test cases of size N. To test specific scenarios and edge cases, we will need a way to define a bin packing problem in a file and load that file into our algorithms.

**Problems with the Problem**

The naive approach has time complexity θ(N2). Optimizing our algorithms to process N elements in θ(N log N) time will be a significant challenge. To test this time complexity fully, we will need to design clever test cases that push the algorithms away from their most efficient states. We will also need to design intelligent methods of verifying this time class experimentally.