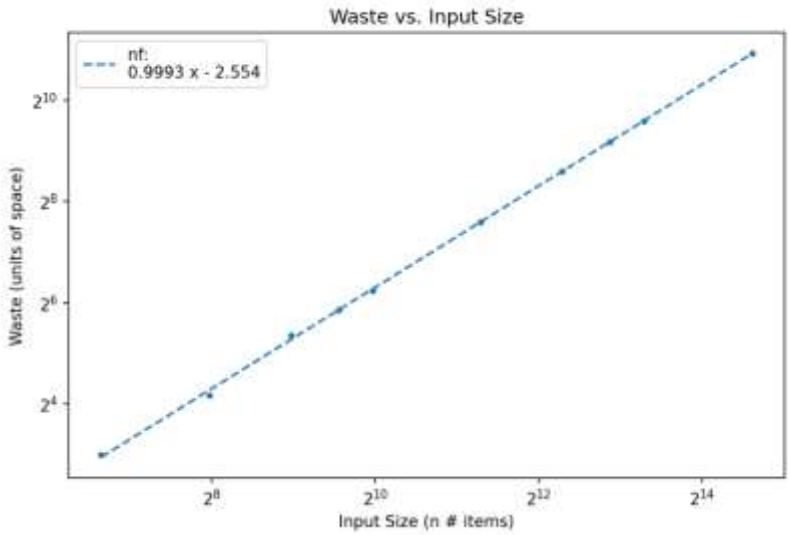
Eric Trinh

Project #2 – Analyzing Bin-Packing Algorithms

In this report write-up, we will be looking at various bin packing algorithms and analyzing how the waste of each algorithm scales with increasing input sizes. The five sorting algorithms implemented include next-fit, first-fit, first-fit-decreasing, best-fit, and best-fit-decreasing.

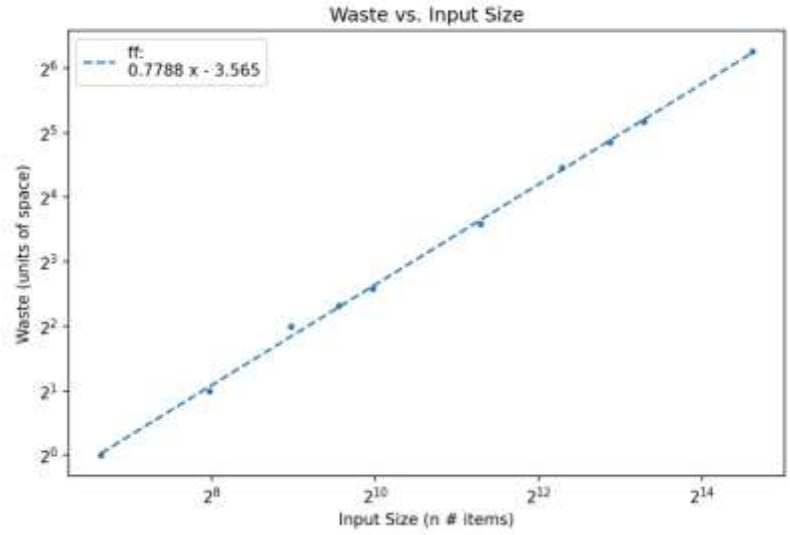
Next-Fit Bin Packing ----------

The next-fit bin packing algorithm looks at the last bin to see if the item fit. If the item fits, then the item is added to the bin, else, a new bin is added, and the item is added into the bin.



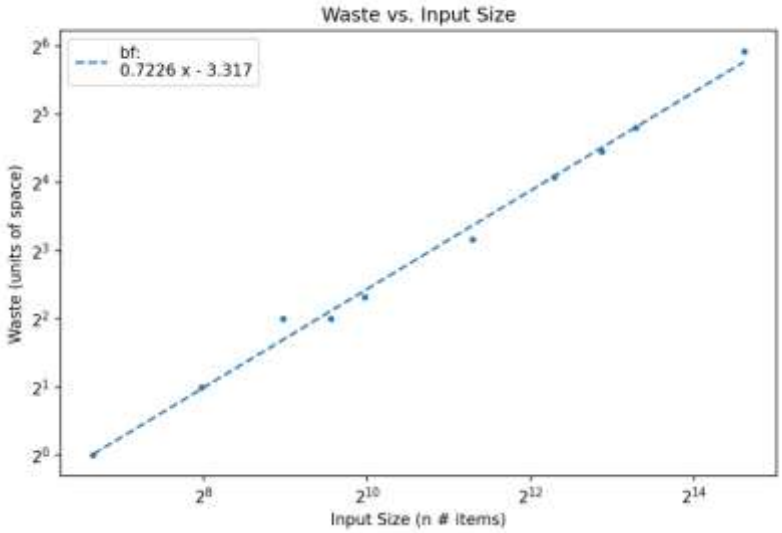
First-Fit Bin Packing ----------

The first-fit bin packing algorithm iterates through all the current bins to find a bin that can store the current item. The item is added to the first bin that is unavailable. If the item does not fit any of the bins, then a new bin is added, and the item is added to the bin.



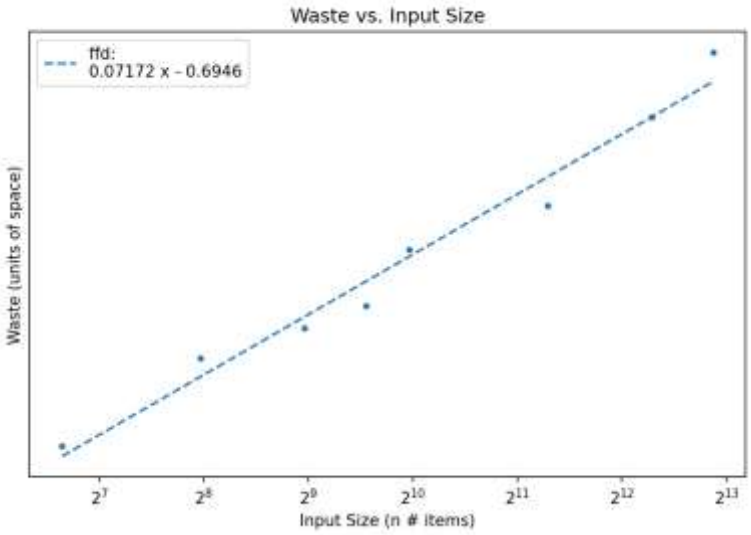
Best-Fit Bin Packing ----------

The best-fit bin packing algorithm looks through the current bins and finds the bin that the item best fits (meaning that the size of the item is closest to the free space left). If the item doesn't any of the bins, a new bin is added, and the item is added to the bin.



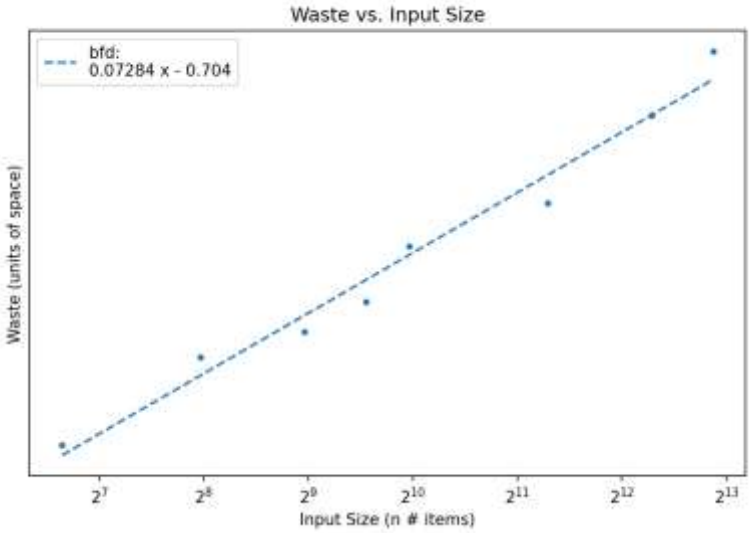
First-Fit-Decreasing Bin Packing ----------

The first-fit-decreasing bin packing algorithm uses the same bin packing algorithm as first-fit except that the input items are sorted from largest to smallest and then the first-fit algorithm is run.



Best-Fit-Decreasing Bin Packing ----------

The best-fit-decreasing bin packing algorithm uses the same bin packing algorithm as best-fit except that the input items are sorted from largest to smallest before the best-fit algorithm is run.



Ranking for Waste Efficiency of Bin Packing Algorithms ----------

Both the decreasing bin packing algorithms are the most efficient regarding waste. This makes sense because by sorting the input items first from largest to smallest, we reduce the chances of large fragmentation occurring because it will be less likely that small item gets placed in a bin with a lot of free space because the smallest items are at the end.

Out of the three main algorithms, best-fit is the most waste efficient because while it runs slower than both next-fit and first-fit, it is less prone fragmentation because the algorithm inserts items to where they best fit. Because the algorithm uses more time to check the bins for the minimum capacity that the item fits, it is more waste efficient but slower. With respect to time complexity, next-fit is the fastest algorithm and best-fit is the slowest out of the three. Conversely, next-fit is the least efficient with waste and best-fit is the most efficient.

This would technically make best-fit-decreasing the most waste efficient out of the five because it checks for best fit, and the items already are sorted. The ranking of the waste efficiency of all the five algorithms are as follows from best to worst: best-fit-decreasing / first-fit-decreasing, best-fit, first-fit, and then next-fit.