

Assessment 3: Bin Packing Problem

Done By:

Wei Ming Edward Ong (14005817)

Byron Lestor (14386344)

GITHUB LINK: [HTTPS://GITHUB.COM/EDDYBOON/ADVANCEDALGORITHMS](https://github.com/EDDYBOON/ADVANCEDALGORITHMS)

Table of Contents

1.	BIN PACKING PROBLEM.....	1
2.	OVERVIEW OF ALGORITHM	1
3.	ASSUMPTIONS MADE	2
4.	TIME AND SPACE COMPLEXITY OF SOLUTION	2
5.	RUNNING THE PROGRAM	2
5.1.	HOW TO RUN	2
5.2.	CLEANING UP	2

1. Bin Packing Problem

The bin packing problem is a combinatorial problem in which the goal is to pack objects of various sizes into a finite number of bins so that the total number of bins used is minimized. Each bin has a fixed capacity, and no bin can contain objects whose combined size exceeds its capacity.

2. Overview of Algorithm

The algorithm undertaken was similar to patience sort but with a twist. The inputs are a list of item sizes and the bin capacity. The items are then sorted from the largest size to the smallest size. An empty list of bins is initiated, and for each item, we will try to place it in the first bin that has enough remaining space. If no such bin exists, we will create a new bin and place the item in it. Trying to fit the item into each bin, starting from the first, ensures the best combination to provide the minimum number of bins.

3. Assumptions Made

- All objects have positive weights.
- All objects have to be fitted into a bin.
- If an object's weight is more than the fixed capacity of a bin, the problem is unsolvable, and -1 will be given since all objects need to be fitted into a bin.

4. Time and Space Complexity of Solution

$O(n+m)$ where n is the length of the items list and m is the length of the list of bins. This is because we have to loop through the list of items at least once and assuming that all items fit the bin's capacity with no space remaining, we have to loop through all the bins to create a new bin at the end. Hence, $O(n+m)$.

5. Running the Program

5.1. How To Run

- Make sure you are in the root directory.
- Run the "make" command.
- Run `./binPackingTest` to execute the tests
- Repeat the steps when changes are made to any file to recompile.

5.2. Cleaning Up

- Run "make clean" to remove the compiled files. This will restore the project to the original state with only the source files.