

CS 325- Homework Assignment 8

Problem: In the bin packing problem, items of different weights (or sizes) must be packed into a finite number of bins each with the capacity C in a way that minimizes the number of bins used. The decision version of the bin packing problem (deciding if objects will fit into $\leq k$ bins) is NP-complete. There is no known polynomial time algorithm to solve the optimization version of the bin packing problem. In this homework you will be examining two greedy approximation algorithms to solve the bin packing problem.

- First-Fit: Put each item as you come to it into the first (earliest opened) bin into which it fits. If there is no available bin then open a new bin.
- First-Fit-Decreasing: First sort the items in decreasing order by size, then use First-Fit on the resulting list.

a) Give pseudo code and the running time for each of the approximation algorithms.

b) Implement in C++. Both algorithms should be in a program called bin.cpp. Your program should read in a text file named bin.txt with multiple test cases as explained below and output to the terminal the number of bins each algorithm calculated for each test case. Your program should run with the command `g++ bin.cpp`

Example bin.txt: The first line is the number of test cases, followed by the capacity of bins for that test case, the number of items and then the weight of each item. You can assume that the weight of an item does not exceed the capacity of a bin for that problem.

```
2
10
6
5 10 2 5 4 4
10
20
4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6
```

Sample output:

```
Test Case 1 First Fit : 4 - First Fit Decreasing : 3
Test Case 2 First Fit : 15 - First Fit Decreasing : 10
```

c) Summarize your results on randomly generated sample input. Which algorithm performs better? How often?

EXTRA CREDIT: An exact solution to the bin packing optimization problem can be found using 0-1 integer programming (IP) see the format on the [Wikipedia page](#). Write a program named binIP.cpp that takes as input the bin.txt file and produces as output the corresponding problems in integer programming format that can be solved using LINDO or Matlab. Compare the optimal solutions from the integer programming solutions to your results using the approximation algorithms. Submit binIP.cpp and the IP files along with a verbal description, running time and README file.