

CSci 3501 Lab 4  
30 points

- Algorithm and implementation: 12 pts
- Setup (input/output) 3 pts
- Documentation, explanation of the algorithm: 5 pts
- Examples of optimal/non-optimal solutions, test data: 5 pts
- Efficiency analysis: 5 pts

Work in pairs

- All lab submissions should be done by canvas. Please be sure to include your group members.
- When working on the lab, please comment your work so that it is clear what the contributions from each person are.
- At the end of the lab each group should submit the results of their in-class work. Please indicate if this is your final submission. Don't forget to answer all the questions below.
- If your submission at the end of the lab time was not final, please submit a final copy before the due time.

---

**Overview and goals**

The goal of the lab is develop and implement an **approximation** algorithm to solve a **Bin Packing Problem**: find a way to pack  $N$  items of different fixed sizes into 3 bins with capacity  $B$  each. The algorithm should try to minimize the amount of unused bin space (or, equivalently, the total unpacked amount).

Example: given  $B=20$  and items of sizes

12, 4, 8, 15, 9, 3, 1, 10

you can pack the bins as [12, 8], [1, 4, 15], and [9, 10], with the leftover item

of size 3.

It is proven that to find the optimal solution, one needs to consider all possibilities of packing which is exponential in  $N$ . However, there are good approximation algorithms that find a solution close to optimal in times polynomial in  $N$ . Your goal is to develop such an approximation algorithm, implement it, test it, and analyze its efficiency.

### Specific Requirements

- Your algorithm must be an approximation algorithm, it is not allowed to check all possibilities.
- Your algorithm does not have to find the optimal solution for all cases but it should find a good solution for most cases.
- The **input** includes the capacity of the bins followed by the number of items  $N$  followed by the items themselves separated by spaces. For instance, the input for the above problem is 20 8 12 4 8 15 9 3 1 10 Here  $B=20$ ,  $N=8$ , and the 8 items follow. See the Scanner class which allows you to read data from the Java console (standard input). Your output should be the list of elements in each bin, the list of the unpacked items, and the amount of unused space. Please make your answers readable.
- You may use any of the algorithms implemented in Java Collections class or any other predefined Java algorithms.  
See <http://docs.oracle.com/javase/tutorial/collections/algorithms/index.html> for a tutorial on the Collections class.

---

### What to submit

- Your program, appropriately documented
- The test data and the results
- A clear explanation of your algorithm in English

- A brief explanation of when your algorithm finds the optimal solution and when it does not. **Give an example** of data for which your algorithm finds a solution that is not optimal. As an extra credit, compute the maximal difference between your solution and the optimal one (in percentage of the total space), show and justify your computations.
- Compute the efficiency of your algorithm in terms of Big-O (by which I mean Big-Theta of the worst case). Clearly explain it. If your program is calling a library method to perform a task or uses a Collections data structure, include efficiency of these operations. If you have questions about these methods and operations, let me know.