CSE 3500 — Algorithms and Complexity — Yufeng Wu — Spring 2016
Programming Assignment ——- Due: 03/25/16 at the end of the day.

In this assignment, you are going to implement two algorithms for determining whether a number is a sum of two numbers in a given list of integer numbers. This is the problem five in HW2. For simplicity, we allow a number to be used twice in forming the sum. For example, if the list contains the number 10, then $20 = 10 + 10$ and thus 20 is considered to a sum of two numbers from the list.

For that homework problem, there are two algorithms:

Brute-force You simply try all possible pairs of numbers to see if a given number matches each sum.

Binary-search You would first sort the list of numbers and then perform binary search on the numbers.

Now implement both algorithms using your favorite language (perhaps Java). Then run your program on the provided data files to compare how the two different algorithms perform on small to large data files. Here are more detailed instructions:

File format After de-compression, you should see three kinds of files: listNumbers-n.txt, listNumbers-n-wsol.txt and listNumbers-n-nsol.txt (here n is the count of numbers in the given list). listNumbers-n.txt contains the given list of integer values (one number per line). listNumbers-n-wsol.txt (for only n=10 is provided): contains a list of numbers (in the 5th column) and how it is obtained (either a sum of two given numbers or just a random number). Here, usually a random number will not be a sum of two numbers (but occasionally it may due to chance and don't worry about this since our purpose here is testing the algorithmic efficiency). listNumbers-n-nsol.txt: a list of 10 numbers (one per line) for which you need to determine whether the number is a sum of two given numbers or not.

Implementation You may use existing Java functions to perform sorting and binary search. If you write sorting and binary search yourself, be careful to make sure they are efficient (e.g. sorting takes $O(nlogn)$ time).

What to submit?

Submit a short report containing the following.

Settings Write down the language you use, the machine (its CPU frequency and memory size) you use for testing your program.

Results Provide a table comparing the two algorithms by listing the average running time for each data size. Then plot the running time in a diagram (with the x-axis being the data size and y-axis being the running time); choose the proper scale to best demonstrate the results.

Conclusion Draw conclusion on why choice of algorithm matters.

Code Attach the source code of your implementation. If it is short enough, you may simply include your code as part of your report.