Tutorial 02 – Task description

Let us assume that you are asked to design and implement an algorithm, which, given a set of integers, identifies all those possible triples of integers the sum of which results to zero (0). You can also assume that your algorithm reads the data from text files such as the file *1Kints.txt*.

- Download both algorithmic implementations, **ThreeSum** and **ThreeSumFast;**
- Download all four files with the test data;
- Complete the missing code for the *ThreeSum* algorithm;
- Run both algorithms with the data from the files **1Kints.txt, 2Kints.txt, 4Kints.txt** and **8Kints.txt (see folder Test data 3-sum)**, respectively, and capture the time being spent for each input data size.
- Using your *padlet link* below, do the following:
    - Study the code and explain the algorithmic strategy being followed;
    - Analyse the performance by contrasting these two algorithmic approaches. The discussion shall be based on an empirical study by measuring the time taken to run the algorithms with different data sizes as of the four data files above.
    - Based on these observations, discuss and suggest the likely classification of the algorithmic performance analysis for *ThreeSum* and *ThreeSumFast* by using one of the seven most common orders of growth classification (see also lecture notes, week 2).

-----------------Specific configuration guidelines for this exercise---

Compilation within a console or terminal:

* Windows version: javac - cp .; stdlib.jar ThreeSum. java

* OS / Linux version: javac - cp .: stdlib.jar ThreeSum. java

Execution within a console or terminal:

* Windows version: java - cp .; stdlib.jar ThreeSum 1 Kints. txt (or 2 Kints. txt, or 4 Kints. txt, or ....)

* OS / Linux version: javac - cp .: stdlib.jar ThreeSum 1 Kints. txt (or 2 Kints. txt, or 4 Kints. txt, or ....)

For NetBeans: Right mouse click on the application main folder, select the option "Set Configuration" and "Customize..." from the drop-down menu. Highlight **Run** and, subsequently, type in and select the file name and the working directory into the text fields **Arguments** and **Working Directory**, respectively.

Notes:

ThreeSumFast is based on an algorithmic strategy, where the input array is $S[0..n-1]$ and has been sorted. 3SUM can be solved faster, on average, by adding numbers pairwise and checking whether the set of input integers contain the integer $-(S[i]+S[j])$ by applying binary search.