FSM FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ	Mühendislik Fakültesi Bilgisayar Mühendisliği Programı **BLM19307E Algorithm Analysis & Design Lab Work**	Adı Soyadı: Notu:

Lab 2: Time Efficiency Comparison

In this lab section, you are required to design an implement an algorithm that finds the count of the pairs that sum to "0" in an array of positive and negative integers assuming that there are no duplicates.

Step1: You are given the main class and some other class files as well as text files that will be used as input to your program.

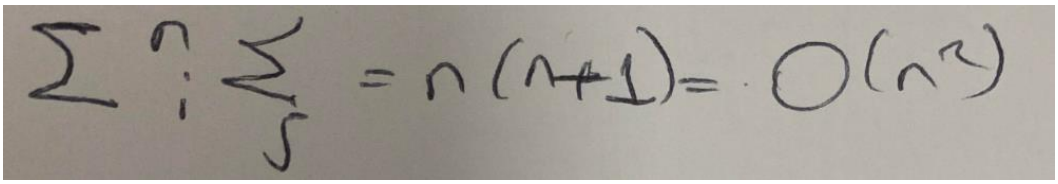
1. First create a new package named "week2_lab"
2. From the course web page, download the code zip file and add the followings to your new package:Lab2.java, StdIn.java, StdOut.java, Stopwatch.java, Solution.java, SolutionFast.java, input_files folder

Step2: In Lab2.java, main method reads in the text file which is given as parameter to the "In" class and creates an array. Then, it calls the "**count**" method of the Solution class. You are required to implement this "count" method which takes in an array and returns an integer which is the count of the distinct pair of numbers whose sum equal to "zero".
Algorithm: (15pts)

```
int n = a.length;
      int counter = 0;
      for (int i = 0; i < n; i++) {
          for (int j = i; j < n; j++) {
              if (a[i] + a[j] == 0) {
                  counter++;
              }
          }
      }


      return counter;
  }
```

Analyze the time efficiency of your algorithm:

- what is the basic operation (**5pts**) Summarization :
- summation for the #of times the basic operation executed (**15pts**) :
- simplify the sum (**5pts**)
-


-
-
- :

Step3: Now uncomment the second part in the main method. Here, another algorithm is used to do the computation. Run both of the algorithms using each of the following inputs and write down the running times in the table below. Then, visualize these using any style of your choice (**25pts**).

input	Solution*	SolutionFast*
1Kints.txt	0.004	0.002
2Kints.txt	0.009	0.003
4Kints.txt	0.023	0.006

8Kints.txt	0.049	0.005
16Kints.txt	0.182	0.007
32Kints.txt	0.64	0.009
1Mints.txt	0.858	0.50

* These times may change depending on the hardware configuration of your computer!

Step4: Answer the following questions:

1. Why does the second algorithm run faster? (**15pt**)

    1. Algoritmada ilk önce sorting işlemi yapılıyor. Dolayısıyla tek tek compare işlemi yapılır. Bst de elemanlar sırayla eklenir.

2. What is the time efficiency class that this algorithm belongs to? Why explain (**20pts**).

    1. Algorithm: O(n*log(n))
    2. Algorithm: O(logn),
    Every time you double the number of nodes in the BST, you only increase the number of steps to the solution by one. To extend this, the nodes give two extra steps four times.