Project: It's About Time!

For this first project, you will create a **linear** search and **binary** search and record the time it takes for searching for various numbers. There is starter code at the end of this document.

What are the requirements for the project?

The following are **required** to earn points for the project:

ArrayList (10%)
Create an ArrayList and add 1,000,000 random numbers that range from 0 - 999,999
Be sure to initialize your Random Seed this way so the values are relatively the same:

```
Random rand = new Random(1);
```

Also, you can use Collections.sort to sort your ArrayList before trying to search for numbers.

Linear Search (25%)
Create a **linear** search method called linearSearch with the method signature below. This method takes the ArrayList of Integer and the number we are searching for and should return the **index** of the number in the array or -1 if not found.

```
public static int linearSearch(ArrayList<Integer> a, int search)
```

Binary Search (50%)
Create a **binary** search method called **binarySearch** with the method signature below. This method takes the ArrayList of Integer and the number we are searching for and should return the **index** of the number in the array or -1 if not found.

```
public static int binarySearch(ArrayList<Integer> a, int search)
```

Report (15%)
In the comments at the top of your code (see Main.java below), record the time for searching for the following numbers for both linear and binary search: 0, 100, 2500, 50000, 100000, 1000000. Is this what was expected? Write a couple sentences about what happened. Be sure to start the clock after you're done initializing the ArrayList. We just want the search times. You can use System.nanoTime() to get the time in nanoseconds. For instance:

```
long startTime = System.nanoTime();
```

How do I submit my work?

Your project must compile! Non-compiling projects will get a 0. When you are done, you must upload your code to the **Assignments** area. This must be code files. Do not take pictures of code or email code.

Projects received 1 minute late are considered late. Start uploading your project at least an hour before the deadline to avoid a point deduction. If there are any issues with uploading your project, you must **email me before the due date**. Email cguida@pace.edu from your @pace.edu email address.

While I check email regularly, **do not expect a response over the weekend or close to deadlines**.
Late projects will have **10 points deducted per day**. Late projects will **not be accepted after 2 days**.

Plagiarism, cheating and other ways you will get a 0 on this project:

Your code must be your own code. You can use code from class or the textbook. Do not use Chegg, CourseHero, ChatGPT or any other websites like these. If you watch a YouTube video or other online resource and put in their code, that's not your code. **That is someone else's code**. You will get a 0. **Do not share your code, or "work on it together". Both** of you will get a 0.

Help, I'm stuck!

Start the projects early, if you are stuck, **reach out to me** cguida@pace.edu, stop by office hours and make use of the **Seidenberg** tutors for help.

Starter Code: Main.java

```
/*
  Your Name:  [YOUR FULL NAME]
  Pace Email: [YOUR PACE EMAIL ADDRESS]

Record the time for the following search:

Linear Search:
          0:
        100:
      2,500:
     50,000:
    100,000:
  1,000,000:


Binary Search:
          0:
        100:
      2,500:
     50,000:
    100,000:
  1,000,000:

Write a couple sentences about what you observed:


*/

class Main {
  public static void main(String[] args) {


  }
}
```