Algorithms 2025/26: Tick 1

A large number of points lie in the 2D Cartesian plane. Your task is to devise and implement an algorithm that identifies which two points are closest together. The points are stored in a plain text file with a comma-separated pair of coordinates on each line, for example:

```
0.3,12.6
10.0,-30
2.3,4
```

Implement your algorithm in Java. Your implementation should be contained in a single Java class which implements the interface below: the *search(String filename)* method is passed the name of the file containing the list of coordinates and should return a string "`a,b c,d`", where (a,b) and (c,d) are the coordinates of the two closest points. If multiple pairs of points are equal-closest, then the return string should contain any one pair.

Your implementation should not require any libraries or external code other than the Java standard library (e.g. `java.util.*` and `java.io.*` classes). I recommend using utility classes in the java.io and/or java.util packages, and the `Double.parseDouble` static function.

To pass Tick 1, your algorithm must be in O(n^2), where n is the number of points in the file.
To earn Tick 1*, your algorithm must be in O(n log n).

Submit a single Java source file containing a class that implements the interface provided. Helper functions are allowed and encouraged to make your code readable! You may write unit tests and a class containing a main(String[]) function in other files but are not asked to submit those.

Submit >> here << You may re-submit if you wish. **The deadline is 12:00 on Mon 9 Feb 2026**.

```
interface Algs202526Tick1 {
  String search(String filename);
}
```

Save this ^^ into Algs202526Tick1.java, then write your solution in Solution.java:

```
public class Solution implements Algs202526Tick1 {...}
```

Dr John Fawcett, Feb 2026