Com S 430
Spring 2018
Homework 4

General instructions

- When submitting modifications of existing code, **please do not reformat any sections of the code that you did not explicitly have to change**. We need to be able to diff your submission against the original to see exactly what was changed.
- Submit an archive on Canvas containing all classes originally posted as hw3.zip.

1) Implement `hw4.subsetsum.findSubsetSumFJ` so that it uses the Fork-Join framework to recursively parallelize the sequential version `findSubsetSum`. The student paper and sample code on this topic can be found on Canvas. In addition, you can find an implementation of merge sort using Fork-Join in the sample code. The tutorial from Oracle might be helpful too. https://docs.oracle.com/javase/tutorial/essential/concurrency/forkjoin.html and this example might also be useful:
https://homes.cs.washington.edu/~djg/teachingMaterials/spac/grossmanSPAC_forkJoinFramework.html

2) Some fun exercises using Java streams. The student papers are available on Canvas. In addition, you'll likely need to refer to:

- the API for `java.util.stream.Stream` and `java.util.stream.IntStream`
- the API for `java.util.stream.Collectors`, which has a number of examples similar to part (b)
- The tutorial on reduction here, which is the example to follow for part (c)
  https://docs.oracle.com/javase/tutorial/collections/streams/reduction.html

There are also two excellent introductory articles provided by Oracle:

- http://www.oracle.com/technetwork/articles/java/ma14-java-se-8-streams-2177646.html
- http://www.oracle.com/technetwork/articles/java/architect-streams-pt2-2227132.html

All the methods in (a) and (b) should be implemented as one liners, using the general form

```
return something.stream()
.blah(...)
.blah(...)
.blah(...)
.blah(...);
```

a) See `hw4.employees.StreamTest`. Write the method

```
public static int[] evensToFront(int[] arr)
```

that, given an integer array, returns a new array with the even numbers in the front and the odd numbers at the end. The relative ordering within the evens and the odds is not changed. (*Tip*: you can us the `.boxed()` method on an `IntStream` to get a stream of `Integer`s, and then use `.sorted()` with a custom comparator.)

b) The remaining stream problems are methods of the class `EmployeeDatabase`. See the javadoc in that class for details. There are some snippets of test code in `EmployeeTest`. There are five methods to implement.

`averageSalaryForDepartment`
`averageSalariesByDepartment`
`listOfEmployeesInProduction`
`getHighestPaidEmployees`
`listEmployeesByPosition`

c) Implement the class **`PositionLister`** (a nested class within **`EmployeeDatabase`**). After this, the method **`listEmployeesByPositionAlt`** should work correctly. Be sure to try it for a parallel stream too!