Exam 3

Logistics

For the final exam, you will work on the programs below by yourself. Please refer to the source code examples found in the `src/homework3` and `src/exam3` folders for hints and guidance. You will receive a separate email with the meeting and demo time on Sunday. During this time, you will demo: (1) your homework assignments (assignment 2 and assignment 3) and (2) the final exam.

Note that **you will be asked questions about the programs.** So please be prepared to answer questions.

Graph Algorithm

A friendship graph can be described with the notation: `<person> <friend 1> <friend 2> ...` . For example:

```
john mike robert james
mike john robert
robert john mike
james john darren
darren james joseph
joseph darren
```

Write a java program that reads in the friendship file and prints out the acquantanceship path between two people.

```
java exam3.graph <friendship file> <person 1> <person 2>
```

For example, if you run the program on the above data set:

```
java exam3.graph friendship.txt mike joseph
```

You might see:

```
mike john james darren joseph
```

If there are no paths between two people, then your program should print `no relationship` .

Tree Algorithm

Write a program that reads in a series of numbers from a file then insert the numbers into a binary tree. The first number is the root node, and subsequent numbers are either pushed to the left or right side of the parent node.

```
5 8 2 4 6
```

Where 5 is the root node. 8 is the right child of 5, 2 is the left child of 5, 4 is the right child of 2, and 6 is the left child of 8.

Your program will also perform an (a) in-order traversal (b) pre-order traversal, and (c) post-order traversal to print out the nodes.

Your program will be executed as followed:

```
java exam3.tree <tree file>
```

The output for the above tree would be:

```
pre-order: 5 2 4 8 6
in-order: 2 4 5 6 8
post-order: 4 2 6 8 5
```

Multithread program

Write a program that reads in words from a text file. The words can be stored in any container of your choice (Tree, Hash, Array, etc).

Your program will create 26 threads. With thread 1 being responsible for the words that begins with the letter a, thread 2 begin responsible for the letter b, etc. Each thread is responsible for tallying up how many words in the collection that starts with the letter for which it is responsible.

Your program will be executed as followed:

```
java exam3.wordcount <text file>
```

The program will print out something as followed:

```
thread id 1: letter a: count 100
thread id 2: letter b: count 120
...
```

Note that the output will not necessarily be printed in any particular order.

Grading

Each program is worth 33 points. More importantly, consider the following requirements

1. First, you will need to complete the program and the program has to work to receive full credit.
2. Second, you must demonstrate comprehension of the code and algorithm. There will be questions on the programs themselves. So merely copying code off the Internet or another student will not help you pass the course.

Good luck and please ask questions if you need clarifications.