CS 2510 Exam 2 – Spring 2012

Name: ______________________________

Student Id (last 4 digits): ______________________________

• Write down the answers in the space provided.
• You may use all syntax that you know from *FunJava* (that is, the parts of Java we have studied in class), although there are several features you will *not* need.
• When defining methods, you do not need to give a complete class definition—just indicate in which class your method definition should be placed.
• For tests you only need to provide the expression that computes the actual value, connecting it with an arrow to the expected value. For example `s.method() -> true` is sufficient.
• Remember that the phrase "design a class" or "design a method" means more than just providing a definition. It means to design them according to the **design recipe**. You are *not* required to provide a method template unless the problem specifically asks for one. However, be prepared to struggle if you choose to skip the template step.
• We will not answer *any* questions during the exam.

Problem	Points	/
A		/ 5
B		/ 5
C		/10
D		/10
Total		/30

Good luck!

34 Points

Problem 1

The Boston power grid consists of a network of connected electric transformers (like the one that burst into flames last night). There are three basic kinds of transformers: those that simply transmit power to another transformer, those that split electricity to two other transformers, and those that are the end points of the power network that don't connect to any further transformers. When a power grid is set up, it's done in such a way that no power is transmitted from any transformer back to itself (there are no cycles in the grid); cycles cause fires. Transformers also carry some data to help identify each point in the grid.

The following class diagram represent power grids:

```
                    +--------------+<-------------------------+
                    |   Trans<T>   |<-----------------------+ |
                    +--------------+<---------------------+ | |
                    | T data       |                      | | |
                    +--------------+                      | | |
                    / \   / \   / \                       | | |
                    ---   ---   ---                       | | |
                     |     |     |                        | | |
       +-------------+     |     +-----------+            | | |
       |                   |                 |            | | |
       |                   |                 |            | | |
+-------------+      +---------------+  +--------------+  | | |
|   End<T>    |      | Link<T>       |  | Couple<T>    |  | | |
+-------------+      +---------------+  +--------------+  | | |
                  +--| Trans<T> conn |  | Trans<T> one |--+ | |
                  |  +---------------+  | Trans<T> two |----+ |
                  |                    +---------------+      |
                  +-------------------------------------------+
```

A. *(5 points)*

Here is a graphical view of a local power grid:

```
            "BOS"
           /     \
          /       \
      "Logan"     "BPD"
      /    \       |
"Seaport" "ICA"   "MFA"
      \    /  \    |
     "Ferry"   "NEU"
                |
             "Ruggles"
```

When a transformer appears above another with a line between them, it means the upper transformer transmits power to the lower one. The strings given show the data each transformer carries.

Translate the above information into examples of data that represent the grid.

B. *(5 points)*

Here is the data representation of another grid:

```
Node<String> t1 = new Terminal<String>("h");
Node<String> t2 = new Terminal<String>("b");
Node<String> t3 = new Split<String>("a", t1, t2);
Node<String> t4 = new Terminal<String>("z");
Node<String> t5 = new Split<String>("c", t4, t3);
Node<String> t6 = new Repeat<String>("j", t3);
Node<String> t7 = new Repeat<String>("d", t6);
Node<String> t8 = new Split<String>("g", t5, t7);
Node<String> t9 = new Terminal<String>("t");
Node<String> t10 = new Split<String>("s", t9, t5);
Node<String> t11 = new Split<String>("k", t10, t8);
```

Translate the above into a graphical representation of the grid similar to that given in part A.

C. *(10 points)*

One measure of the power grids robustness is the number of paths between two transformers.

Design the method `countPaths` that computes the number of paths from this transformer to a tranformer containing the given data.

D. *(10 points)*

The efficiency of a power grid can be determined by the longest path that power is transfered on.

Design the method `maxLength` that finds the length of the longest connection from this node to any end transformer.