In this programming assignment you will start writing the software that goes inside a router. This will not be real, since real routing software would not be written in Java, but it should basically work.

1. The first part of this assignment is to write a class to do forwarding. This has two major parts, the construction and modification of a forwarding table, and determining the port on which to forward a packet with a specified address. The forwarding will be done on the basis of longest matching prefix. The interface `Forwarder` defines the methods that you will need to write.

   (a) Write a class called `SimpleForwarder` that implements `Forwarder`. You will probably want some private helper methods in this class. It can be extremely simple, as the name suggests, and just keep a list of all the prefix entries, searching linearly through that list looking for the longest prefix match when asked to determine a port.

   (b) Write a class called `TreeForwarder` that does the same job somewhat more intelligently. The previous class has the nasty habit of searching the entire table every time. In this version I want you to build a binary tree structure. At the root is the prefix 0/0 (i.e., everything). The root’s left child will represent the prefix 0/1, and the right child the prefix 128/1. Similarly, the four nodes at the next level of the tree will represent prefixes 0/2, 64/2, 128/2, and 192/2, and the next level will be 0/3, 32/3, 64/3, ... Clearly this tree could be up to 32 levels high.

   Each of the nodes will carry a port number. (It doesn’t need to keep track of the prefix itself because that is implicit in the structure of the tree.) A node will only be present if there is a prefix table entry at or below it.

   When forwarding, the job is easy. Start at the root of the tree, and the most significant bit of the address. Go down left or right depending on whether the bit is 0 or 1. Look at the next bit and repeat. At the point there is no child you have found the correct node and can simply return its port number.

   A diagram or two may help. In the following the prefixes (e.g., 0/0) are only present to help you understand the diagram. The only values actually maintained in each node are the port number.

At the beginning we only have the default routing, 0/0 goes out port 0.

0/0 to 0

If we add a prefix 64/2 forwarded on port 1, we will end up with the following tree:

```
  0/0 to 0
     /   \
 0/1 to 0
     /   \
64/2 to 1
```

Now add 128/3 forwarded on port 2: