EE 453/717: Advanced Computing for Electrical Engineers

Indian Institute of Technology Bombay
Autumn 2010

Assignment 1 : **3 points** **Due date**: August 9, 2010

1. Implement the following algorithms for primality testing as C++ programs

   (a) Given a positive integer $p > 1$, divide p by the integers $2, 3, \ldots, \lfloor \sqrt{p} \rfloor$ in order. If the remainder is zero for any of the division operations, p is composite. Otherwise p is prime.

   (b) Given a positive integer $p > 1$, divide p by the integers $2, 3$, and $6k \pm 1 \leq \lfloor \sqrt{p} \rfloor$ $(k = 1, 2, \ldots)$ in order. If the remainder is zero for any of the division operations, p is composite. Otherwise p is prime.

   [1 point]

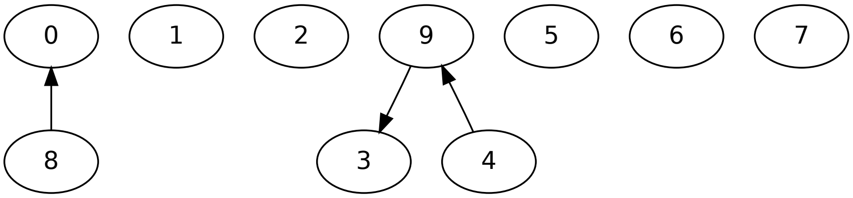

Figure 1: Graph generated by the `dot` tool

2. Graphviz (`http://www.graphviz.org`) is a graphical visualization package which can be used to draw graphs which are specified in simple text files. The `dot` tool in graphviz is used to draw directed graphs. For example, the following file `qf.dot` can be processed by the `dot` tool to generate the graph shown in Figure 1. The command used is `dot -Tpdf qf.dot -o qf.pdf`

```
digraph qf
{
    0;
    1;
    2;
    9 -> 3;
    9 -> 4[dir =back];
    5;
    6;
    7;
    0 -> 8[dir =back];
    9;
}
```

(a) Write a C++ function `drawQuickFindGraph` which generates text files which can be consumed by the `dot` tool to generate graphs corresponding to the *quick find* algorithm discussed in class. [1 point]

(b) Write a C++ function `drawQuickUnionGraph` which generates text files which can be consumed by the `dot` tool to generate graphs corresponding to the *quick union* algorithm discussed in class. [1 point]

You can use the *quick find* and *quick union* implementations from `http://www.cs.princeton.edu/~rs/Algs3.cxx1-4/code.txt`.