CSC 212—Algorithms and Complexity—Graph searching I .

Practical 4 Term 4 — 28 August 2015

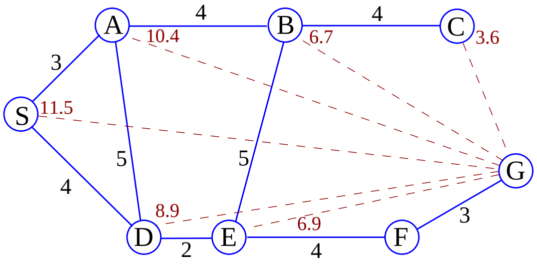
Warm up by writing a procedure that prints the binary digits of any given floating point number that is either a 32-bit `float` or a 64-bit `double`.

Write code that will read in a configuration for some graphs and do depth-first search (DFS), breadth-first search (BFS) and DFS with hill-climbing to determine the path to the goal state starting at a given start state. The diagram below depicts a graph.

1. Work in directory `04practical` again and add the new code. Your code must print the binary bits for `float`s and `double`s of the numbers 1.0, 2.0, 4.0, 57.0, 1023.0, 1024.0, 2.0^{127} and their negatives. Your output must show the original number and the binary representation of the positive and negative numbers for each number. Print the values of the biggest positive and negative floating point numbers.

   The code that produces the "binary string" must be in a form of a method that can print all the relevant bits of the 32- and 64-bit numbers.

2. In a directory `44practical`:



   This graph can be specified using the following data.

```
1  "Weighted graph with estimated distance left to goal"
0                      // start node
7                      // goal node
0: 1,3,10.4; 4,4,8.9$  // S: A  D
1: 2,4,6.7;  4,5,8.9; 0,7,11.5$   // A: B D S
2: 1,4,10.4; 5,5,6.9; 3,4,3.6$    // B: A E C
3: 2,4,6.7$                       // C: B
4: 1,5,10.4; 5,2,6.9; 0,4,11.5$   // D: A E B S
5: 2,5,6.7;  4,2,8.9; 6,4,3.0$    // E: B D F
6: 5,4,6.9;  7,3,0.0$             // F: E G
7: 6,3,3.0$                       // G: F
#
#
```

   (a) **The file system**: Put today's work in your `44practical` directory that contains a correct, appropriate working `Makefile`.

   (b) Your data must be read from a file that contains the specifications of the graph as shown above. A copy of a file called `graphsearch.text` is available in the usual place. Please read directly from this file and *not* from a copy of it stored in you own file system.

(c) Write two separate methods. The start node is S and the goal node is G. Your code must find the goal node initiating its search at the start node.

  i. DFS for graph searching.
  ii. BFS for graph searching.

  In each case display the path to the goal G and give the number of nodes processed in order to find the goal node.

(d) Hand by entering `cd; make submit` in the command-line interface.