CSC 340 DESIGN AND ANALYSIS OF ALGORITHMS
HOMEWORK PROJECT 1

DR. GODFREY MUGANDA

1. The Assignment

Write a program that determines the number of different directed paths from a vertex s to a vertex t in a directed acyclic graph. The program should use an algorithm based on depth-first-search and work in linear time.

Write two versions of your solution, one using a recursive depth-first search, and the other based on a non-recursive depth-first search.

2. Input

Input will be provided to the program in files that contain adjacency lists of directed acyclic graphs. Each file starts of with an integer N, the number of vertices in the graph, followed by N data sets, with each data set representing a vertex v , followed by an integer M representing the out-degree of v, followed by the M neighbors reachable from v.

Vertices are strings that have to be internally represented as integers, and translated back to strings for the purpose of doing output. Code for reading the files and building the adjacency lists will be provided to you, along with an example program that implements a breadth-first traversal of a graph.

Here are three example inputs. First, a DAG with 10 vertices:

```
10
a 3 b f i
f 2 b g
g 2 b e
b 2 c h
c 1 d
h 2 i j
j 2 i d
i 2 c d
d 1 e
e 0
```

Here is a DAG with 9 vertices:

```
9
a 2 b e
b 2 c h
c 0
e 3 d f g
f 2 d i
```

```
g 1 h
h 1 i
d 0
i 0
```

And here is a DAG with 6 vertices:

```
6
a 3 b c d
d 2 c f
b 3 c d g
c 0
f 1 c
g 1 c
```

Here is a sample run that uses the first and third data sets:

```
Is there another graph to process? (Y/N)y

Vertex name to integer map:
{a=0, b=1, c=6, d=8, e=5, f=2, g=4, h=7, i=3, j=9}
Vertex integer to string names map:
[a, b, f, i, g, e, c, h, d, j]

Adjacency list for the graph in string names form:
a : [b, f, i]
b : [c, h]
f : [b, g]
i : [c, d]
g : [b, e]
e : []
c : [d]
h : [i, j]
d : [e]
j : [i, d]
Enter a source vertex, blank line to quit: a
Enter a destination vertex, blank line to quit: e
Number of Paths  from a to e found by non-recursive method.
a : 21
b : 6
f : 13
i : 2
g : 7
e : 1
c : 1
h : 5
d : 1
j : 3

Number of Paths  from a to e found by recursive method.
a : 21
b : 6
f : 13
i : 2
```

```
g : 7
e : 1
c : 1
h : 5
d : 1
j : 3

Enter a source vertex, blank line to quit:
Is there another graph to process? (Y/N)y

Vertex name to integer map:
{a=0, b=1, c=2, d=3, f=4, g=5}
Vertex integer to string names map:
[a, b, c, d, f, g]

Adjacency list for the graph in string names form:
a : [b, c, d]
b : [c, d, g]
c : []
d : [c, f]
f : [c]
g : [c]
Enter a source vertex, blank line to quit: a
Enter a destination vertex, blank line to quit: c
Number of Paths  from a to c found by non-recursive method.
a : 7
b : 4
c : 1
d : 2
f : 1
g : 1

Number of Paths  from a to c found by recursive method.
a : 7
b : 4
c : 1
d : 2
f : 1
g : 1

Enter a source vertex, blank line to quit:
Is there another graph to process? (Y/N)n
Goodbye
```

3. Further Requirements

All programming projects must be done in Java, using the Netbeans IDE, unless otherwise specified. You must submit by uploading the entire Netbeans project folder onto the K-drive. Work that is not submitted according to this requirement will not be graded.

I will usually provide starter code that performs input and output, to allow you to focus all your time on the design and implementation of the algorithm itself.

Starter code will be in Java 8/ Netbeans 8, but you may use a latter version of Java / Netbeans if you like.

4. Due Date

February 15 at midnight. You should get started on it right away because Project 2 will be assigned before the due date for this one.