COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Technology, Palakkad

CS2130: Data Structures and Algorithms Lab

Lab 8 (Breadth First Search)

28 Sep, 2018

Time: 3 hrs

General Instructions

- You are expected to follow good programming practices (refer to the book titled *C How to Program* by Dietel and Dietel). Your programs should handle all kinds of test cases - positive, negative, corner cases/boundary conditions, good input, bad input and are never expected to crash on any input.
- You are also expected to provided a high level description of the functions used in your programs. There is no restriction on the format except that it should be concise and precise. Further, the running time and additional space (excluding the space for the arguments) used by the functions should be clearly stated and justified.
- **Please note that poor programming practices and improper documentation can attract a penalty of up to 20% of the marks allocated to a question.**

Each of the following programs take a directed graph G on n vertices and m edges as input where $n \geq 1$. The vertex set of G is $\{0, 1, \ldots, n-1\}$ and an edge directed from vertex i to vertex j is denoted by (i, j). The input file will contain $m+1$ lines with the first line containing the number n. Each of the subsequent m lines will contain two non-negative integers i and j, separated by a space, denoting edge (i, j) of G.

1. (30 points) Write a program to find the strongly connected components (SCCs) of a directed graph G using the DFS algorithm implemented in the previous assignment. If there are ℓ SCCs in G, then the output consists of ℓ lines with the ith line containing the vertices of the ith SCC separated by \$ in the format given below. Further, for each $1 \leq i \leq \ell - 1$ and $\ell \geq j > i$, the maximum finish time of a vertex in the ith SCC should be larger than the maximum finish time of a vertex in the jth SCC. Here, the finish times refer to the finish times of the vertices obtained during DFS on G^R. Recall that G^R is the reverse of G with $V(G^R) = V(G)$ and $E(G^R) = \{(v, u) : u, v \in V(G), (u, v) \in E(G)\}$. For example consider the following sample input graph G.

```
7
0 4
4 0
1 2
2 3
3 1
5 6
6 5
```

Continued on next page. . .

On executing the DFS algorithm on G^R, the ordering of the vertices in the decreasing order of their finish times is $5, 6, 1, 3, 2, 0, 4$. Therefore, the first line of the output should contain the vertices of the SCC of G that contains vertex 5. This is the set $\{5, 6\}$ displayed as 5\$6\$. The next line should contain the vertices of the SCC of G that contains the vertex that has the largest finish time among the vertices in $V(G) \setminus \{5, 6\}$. Thus, the second line contains the vertices of the SCC of G that contains vertex 1. This is the set $\{1, 2, 3\}$ displayed as 1\$2\$3\$. The next line should contain the vertices of the SCC of G that contains the vertex that has the largest finish time among the vertices in $V(G) \setminus \{5, 6, 1, 2, 3\}$. So, the third line contains the vertices of the SCC of G that contains vertex 0. This is the set $\{0, 4\}$ displayed as 0\$4\$. Proceeding in this fashion until all SCCs of G are listed, we get the following output.

```
5$6$
1$2$3$
0$4$
```

2. (30 points) Write a program to implement the Breadth First Search algorithm on directed graphs. The output is the set of edges of the BFS tree F rooted at vertex 0 of the input graph G. At any point in the execution of the algorithm, if there are multiple choices for a vertex to be picked, then pick the one with least identifier value. If F consists of ℓ edges then the output file should consists of ℓ lines with each line denoting an edge of F. Each such line should be of the form (i, j) where (i, j) is an edge in F. Each such line should be of the form (i, j) where (i, j) is an edge in F. Further, for each $0 \leq i \leq n-1$, if F has ℓ_i edges from vertex i, then Lines $(i \cdot \ell_{i-1} + 1)$ to $(i \cdot \ell_{i-1} + \ell_i)$ of the output should be these edges displayed in the increasing order of their second endpoints. That is, if (i, k) and (i, j) are edges in F with $k > j$, then (i, j) should be displayed before (i, k).

```
Sample Input:
6
0 2
0 1
2 1
4 5
3 2
1 3
3 4
Expected Output:
(0,1)
(0,2)
(1,3)
(3,4)
(4,5)
```

3. (40 points) Modify the above program to compute the distance between every (ordered) pair of vertices of a given directed graph on n vertices. The output consists of n^2 lines with the distance from vertex i to vertex j displayed in the $(i * n + (j + 1))$th line in the format given below.

Question 3 continues on the next page. . .

```
Sample Input:
6
0 2
0 1
2 1
4 5
3 2
1 3
3 4
Expected Output:
dist(0,0)=0
dist(0,1)=1
dist(0,2)=1
dist(0,3)=2
dist(0,4)=3
dist(0,5)=4
dist(1,0)=-1
dist(1,1)=0
dist(1,2)=2
dist(1,3)=1
dist(1,4)=2
dist(1,5)=3
dist(2,0)=-1
dist(2,1)=1
dist(2,2)=0
dist(2,3)=2
dist(2,4)=3
dist(2,5)=4
dist(3,0)=-1
dist(3,1)=2
dist(3,2)=1
dist(3,3)=0
dist(3,4)=1
dist(3,5)=2
dist(4,0)=-1
dist(4,1)=-1
dist(4,2)=-1
dist(4,3)=-1
dist(4,4)=0
dist(4,5)=1
dist(5,0)=-1
dist(5,1)=-1
dist(5,2)=-1
dist(5,3)=-1
dist(5,4)=-1
dist(5,5)=0
```