Lab06-Graph Exploration

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Mingran Peng.

∗ Name:Lynn Xiao Student ID:_____ Email: _____

1. Given a graph, find the number of Strongly Connected Components in the graph.

   (a) Complete the implementation in the provided C/C++ source code. Notice that in the source code there will be more detailed explanation.(The source code *SCC.cpp* is attached on the course webpage.)

   (b) Use the *Gephi* to draw the graph. If you think the data provided is not beautiful, you can generate your own data. Notice that result of *Gephi* will be taken into consideration of Best Lab.

2. Remember the lemma introduced in the course: : $\forall u, v \in V$, intervals $[PRE(u), POST(u)]$, $[PRE(v), POST(v)]$ are either disjoint or one is contained within the other.

   Prove the lemma.

   Proof.
   If $(u, v) \notin$ E, we can easily know that intervals $[PRE(u), POST(u)]$ and $[PRE(v), POST(v)]$ are disjoint.

   If $(u, v) \in$ E, assume that we first visit v.

   If u is not visited before, then $PRE(u)>PRE(v)$ and $POST(v)>PRE(u)$ since we call $EXPLORE(G, u)$ in $EXPLORE(G, v)$. So $[PRE(u), POST(u)]$ is contained within

   $[PRE(v), POST(v)]$.

   We can get the same conclusion when u is visited first. So the lemma is proved. □

3. Consider there is a network consists n computers. For some pairs of computers, a wire exists in the pair, which means these two computers can communicate with delay t.

   Assume that computer s wants to issue a message to computer t, we want to know the minimum time needed to send this message.

   You need to provide the pseudo code and analyze the time complexity.

   Proof.
   We can use BFS to solve this problem, which is shown in Alg.1.

□

Algorithm 1: Calculate least delay by BFS

Input: Numbers of computers n, computer s and t, Graph $G(V, E)$, delay time T

Output: The minimum time needed to send message from computer s to t

```
1  for u ∈ v do
2      DIST(u) ← ∞
3  DIST(s) ← 0
4  Q ← [s] (Q is a queue)
5  while Q is not empty do
6      u = EJECT(Q)
7      for (u, v) ∈ E do
8          if DIST(v) = ∞ then
9              INJECT(Q, v)
10             DIST(v) ← DIST(u) + 1 if v=t then
11                 return T × DIST(v)
```

Time complexity:
(i) Best case: $O(1)$, if $(s, t) \in E$.
(ii) Worst case: $O(|V| + |E|)$, if every edge is visited.

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.