EECS 340: Assignment 7

Shaochen (Henry) ZHONG, `sxz517`
Yuhui ZHANG, `yxz2052`

Due and submitted on 04/27/2020
EECS 340, Dr. Koyutürk

Problem 1

(a)

Set the each team as a node (vertex) in graph G and make every node fully connected (representing the fact that all teams have played against each other). If one team A beats another team B, we will have a directed edge of $A \rightarrow B$ betwen the nodes representing A and B.

Algorithm 1 DFS(G)

```
procedure DFS(G)
    for each u ∈ G.V do:
        u.color = WHITE
        u.z = 0
    for each u ∈ G.V do:
        if u.color == WHITE then
            DFS-GAME-VISIT(G, u)
```

Algorithm 2 DFS-Game-Visit(G, u)

```
procedure DFS-GAME-VISIT(G, u)
    z = NULL
    u.color = GRAY
    for each v ∈ Adj(u) do:
        if v.color == WHITE then
            z = v.rank
            z = min(z, DFS-GAME-VISIT(G, v))
    u.color = BLACK
    if z == NULL then
        u.z = 0
        z = u.rank
    else
        u.z = z
    return z
```

(b)

Justification for runtime The algorithm is $O(m+n)$ as DFS will first traverse every node, which means it will at least be $O(n)$. In addition, since the graph G is implemented using adjacency list, for each node we will have to traverse through all its adjacent edges – where such number can be at most m (total number of edges in G) for a single node. Thus, the total time complexity is $O(m+n)$.

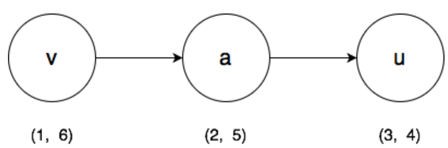
Justification for correctness The algorithm basically performs a DFS travesal on the graph G and set the node `u` with smallest `rank` value as `z` to all of node `u`'s ancestor nodes. Due to the nature of DFS, node `u`'s ancestor nodes can legally have `u.rank` as their domination factor, as being `u`'s ancestors imply the fact these teams have won against team `u`. Also in DFS-Game-Visit we will not set a node's `.z` value unless such node is marked as `BLACK` (which means all of its decedents have been explored). Combine these two observations, each node has explored all of its decendents nodes, and the node's decendent node with the smallest `rank` will be assigned as all its ancestor nodes' `.z` value. The algorithm is a perfect mimic of the game logic of domination factor and guaranteed to be correct.

Problem 2

Note the tuple below each node represents their `(start-time, finish-time)`.

(a)

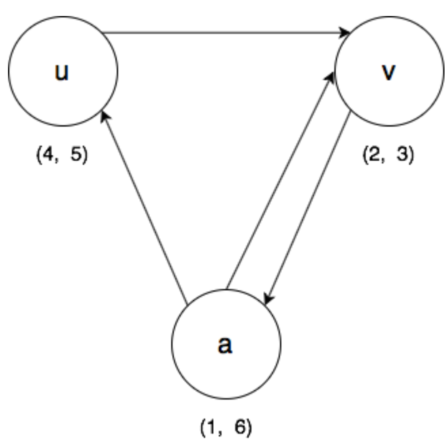
Disprove with conterexample:



It is shown that $v.f > u.f$ as $6 > 4$, however there is no circle between uv. Thus, this conterexample disproves the statement.

(b)

Disprove with conterexample:



It is shown that there is a path from v to u as $v \to a \to u$, however uv is still a cross edge as $v.d < v.f < u.d < u.f$ $(2 < 3 < 4 < 5)$. Thus, this conterexample disproves the statement.

Problem 4

Shaochen (Henry) ZHONG's proof of course evalution

Spring 2020

When your evaluation is submitted, only your sequence of responses and written comments will be reported, without any additional personal identifying information.

EECS 340: Algorithms (110/4975)

Your responses were saved.

Please choose a course to evaluate.

Evaluation	Evaluation period	Already responded?
[redacted]		
EECS 340: Algorithms (110/4975)	Apr 13 - 11:59 PM May 07	Yes
[redacted]		

Yuhui ZHANG's proof of course evalution

Spring 2020

When your evaluation is submitted, only your sequence of responses and written comments will be reported, without any additional personal identifying information.

EECS 454: Analysis of Algorithms (100/4973)

Your responses were saved.

Please choose a course to evaluate.