# Quiz #2
# Introduction to Algorithms/Algorithms 1
# 600.363/463

Thursday, April 3rd, 9:00-10:15am

Name:

Class (circle one): 363 463

**Ethics Statement**

I agree to complete this exam without unauthorized assistance from any person, materials, or device.

Signature Date

# 1 Problem 1 (20 points)

Give a definition of minimum spanning tree.

# 2 Problem 2 (75 points)

## 2.1 (25 points– 15 for part (a), 10 for part (b)

Let $G = (V, E)$ be a directed cycle, given by $V = \{v_1, v_2, \ldots, v_n\}$, and $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}$ (that is, for each $i \in \{1, 2, \ldots, n-1\}$, $(v_i, v_{i+1}) \in E$, and $(v_n, v_1) \in E$.

(a) State whether or not the following claim is true and give a brief explanation as to why: No matter what node $v \in V$ we start DFS at, the shortest paths from $v$ to every other node in the DFS tree are exactly the shortest paths from $v$ to every other node in $G$. You **do not** need to give a formal proof here. Just give a two- or three-sentence sketch of the proof or give a counter-example.

(b) Suppose we make a new graph $G'$ from $G$ by reversing a single edge in $G$, so that our new graph is given by $G' = (V, E')$ where $E' = \big(E \cup \{(v_{i+1}, v_i)\}\big) \setminus \{(v_i, v_{i+1})\}$. State whether or not the following claim is true, and give a brief explanation as to why: No matter what node $v \in V$ we start DFS at, the shortest paths from $v$ to every other node in the DFS tree are exactly the shortest paths from $v$ to every other node in $G$. You **do not** need to give a formal proof here. Just give a two- or three-sentence sketch of the proof or give a counter-example.

### 2.2 (25 points)

Let us say that the Bellman-Ford algorithm *pseudo-terminates* if after the first iteration is finished, no more values change for the remainder of the algorithm. (i.e., no node $v$ has its $v.d$ or $v.\pi$ attributes updated during iterations 2 through $|V|-1$ of the outer for-loop). That is, Bellman-Ford pseudo-terminates on a graph $G = (V, E)$ if after the first iteration of the for loop, we have already found all of the shortest paths.

(a) Suppose that we have a directed path on $n$ nodes, with unit edge weights. Let $v$ denote the unique node in this path that has no predecessors. If we perform Bellman-Ford, and consider the vertices in $G$ in topological order, will Bellman-Ford pseudo-terminate? Explain. You **do not** need to give a rigorous proof, here. A brief explanation and/or counter-example will suffice.

(b) As before, suppose that we have a directed path on $n$ nodes, with unit edge weights and let $v$ denote the unique node in this path that has no predecessors. If we perform Bellman-Ford, and $v$ is *not* the first node considered in the outer for-loop, will Bellman-Ford pseudo-terminate? Again, you need not give a rigorous proof. Explain. You **do not** need to give a rigorous proof, here. A brief explanation and/or counter-example will suffice.

### 2.3 (25 points)

Let $G = (V, E)$ be a connected, undirected graph with non-negative edge weights. Prove or disprove the following claim: if edge $e \in E$ is the unique smallest-weight edge in graph $G$, then $e$ must be included in any minimum spanning tree of $G$.

# 3 Problem 3 (55 points)

Given a weighted directed graph $G = (V, E)$ with non-negative edge weights, call node $w \in V$ a *critical* node for nodes $u \in V$ and $v \in V$ if for all shortest paths $P$ from $u$ to $v$ in $G$, $P$ visits node $w$. Give a polynomial-time algorithm that checks whether or not there is a critical node for nodes $u$ and $v$ in graph $G$. If one or more critical nodes exist for nodes $u$ and $v$, your algorithm should return one. If no critical node exists for nodes $u$ and $v$, return `None`. Prove the correctness of your algorithm and prove that it runs in polynomial time.