First name: __________________________

Last name: __________________________

UCR student ID: ____________________

- This quiz is closed book, closed notes and 30 minutes long
- Read the questions carefully
- No electronic equipment allowed (cell phones, tablets, computers, . . .)
- Write legibly and try to be brief and to the point; what can’t be read will not be graded
- No code: use pseudocode or English to describe your algorithms
- Always remember to analyze the time complexity of your solution
- If you have a question about the meaning of a question, raise your hand
Problem 1. (42 points: 6 points if correct, 3 if unanswered, 0 if wrong)
Mark by true or false each of the following (no need to prove).

\[ \sqrt{n} \log_2 n^3 \in O(n \log_2 n) \quad \square \text{True} \quad \square \text{False} \]

\[ 9^{\log_3 n} \in \Omega(n^2 \log_2 n) \quad \square \text{True} \quad \square \text{False} \]

\[ \log_2 2^n \in \Theta(\sqrt{n} \log_3 3^\sqrt{n}) \quad \square \text{True} \quad \square \text{False} \]

The following questions are on graphs; assume that \( n = |V| \) is the number of vertices, and \( m = |E| \) is the number of edges; DFS is “depth first search”; BFS is “breadth first search”; in DFS/BFS the the set of edges visited during the execution of these algorithms are called tree or discovery edges; non-tree edges are the others (also called back edges in DFS, cross edges in BFS).

Given the spanning tree \( T \) formed by the discovery (tree) edges of a BFS traversal of a connected undirected graph \( G \) started from node \( s \), for each vertex \( v \), the path on tree \( T \) is the shortest path between \( s \) and \( v \) \quad \square \text{True} \quad \square \text{False} \]

An edge \( e \) whose removal disconnects the graph is called a bridge; if BFS is run on a connected undirected graph \( G \), it is a possible for a bridge in \( G \) to be a cross (non-tree) edge \quad \square \text{True} \quad \square \text{False} \]

For a connected undirected graph \( G \), the absence of back (non-tree) edges with respect to a DFS tree implies that \( G \) is acyclic \quad \square \text{True} \quad \square \text{False} \]

If one runs a DFS on a connected undirected graph, the number of back (non-tree) edges is exactly \( m - n + 1 \) \quad \square \text{True} \quad \square \text{False} \]
Problem 2. (24 points: 8 points each)

For each of the concepts listed below write a precise (possibly formal) definition. Do not explain or comment about the corresponding algorithm, if any.

1. topological ordering of a directed acyclic graph \( G = (V, E) \)

2. strongly connected component in a directed graph \( G = (V, E) \)

3. spanning tree of an undirected graph \( G = (V, E) \)
Problem 3. (34 points)

Suppose you are given an array $A = \{a_1, a_2, \ldots, a_n\}$ of $n$ distinct integers. You are told that the sequence of values $a_1, a_2, \ldots, a_n$ is unimodal, that is for some index $p \in [1, n]$, the values in the array increase up to position $p$ in $A$, and then decrease the remainder of the way until position $n$. Give an algorithm to find the position $p$ in $O(\log n)$ time. You can assume $n$ to be a power of 2. Explain why your algorithm runs in $O(\log n)$ time.