• This quiz is closed book, closed notes and 35 minutes long
• Read the questions carefully
• No electronic equipment allowed (cell phones, PDAs, computers, . . .)
• Write legibly. What can’t be read will not be graded
• 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. (45 points: 5 points if correct, 2.5 if unanswered, 0 if wrong)
Mark by true or false each of the following (no need to prove).

4^{\log_2 n} \in \Omega(n^2 \log n) \quad \square True \quad \square False

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

\sqrt{n} \log_3 n^2 \in O(n \log_3 n) \quad \square True \quad \square False

HEAP-SORT runs in \( O(n) \) time \quad \square True \quad \square 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)

An undirected complete graph with \( n \) nodes has exactly \( n(n - 1)/2 \) edges \quad \square True \quad \square False

Given the spanning tree \( T \) formed by the discovery (tree) edges of a DFS 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 True \quad \square False

An edge \( e \) whose removal disconnects a graph is called a bridge; if DFS is run on a connected undirected graph \( G \), every bridge in \( G \) is a discovery (tree) edge in the DFS tree \quad \square True \quad \square False

For a connected undirected graph \( G \), the presence of a back (non-tree) edge in any DFS visit of \( G \) implies that \( G \) has a cycle \quad \square True \quad \square False

If one runs a BFS on a connected undirected graph, the number of cross (non-tree) edges is exactly \( m - n + 1 \) \quad \square True \quad \square False
Problem 2. (25 points: 5 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. transitive closure of an undirected graph \( G = (V, E) \)

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

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

4. cycle in an undirected graph \( G = (V, E) \)

5. binary heap
Problem 3. (30 points)

You are given an unsorted array $A[1\ldots n]$ of $n$ distinct integers. We say that $A[i]$ is a
local maximum if $A[i]$ is bigger than its neighbors, that is, $A[i] > A[i-1]$ (if $i \neq 1$) and
$A[i] > A[i+1]$ (if $i \neq n$). For instance in $A = \{3, 4, 1, 2, 6, 0, 8, 9\}$, 4 is a local maximum, 6
is a local maximum, and 9 is a local maximum. Give an $O(\log n)$-time algorithm to find any
of the local maximum in $A$. If there is more than one local maximum in $A$, we are OK with
any one of them. Explain briefly how the algorithm works, and why it runs on $O(\log n)$ time.
You can assume that $n$ is a power of two. Hint: If $A[i]$ is not a local maximum because
$A[i] < A[i+1]$, then must there be some local maximum $A[j]$ with $j > i$?