1. Choose all the correct statements. [6 points]

- ✓ A. A state-diagram can have zero or more final states, but exactly one start state.
- □ B. There can be no transitions out of a final state in a state-diagram.
- □ C. If a state-diagram is deterministic, and it has two states A and B, then there cannot be two different edges between them, both directed from A to B.
- □ D. None of the above.

2. The problem of graph 3-colorability, 3COL, is \textbf{NP}-Complete. Which of the following statements are known to be implied by this? [6 points]

- □ A. If some problem in \textbf{NP} is in \textbf{P}, then 3COL ∈ \textbf{P}.
- ✓ B. If 3COL ∈ \textbf{P}, then every problem in \textbf{NP} is in \textbf{P}.
- ✓ C. If every problem in \textbf{NP} is in \textbf{P}, then 3COL ∈ \textbf{P}.
- □ D. None of the above.

3. Consider a recursive sorting algorithm \textsc{RSort} that takes \((A, n)\) where \(A\) is an array of size at least \(n\), and sorts the first \(n\) positions of \(A\). (So to sort an entire array using \textsc{RSort}, we can set \(n\) to be the size of the array.)

On input \((A, n)\), \textsc{RSort} works as follows. If \(n = 1\), \textsc{RSort} returns \(A\) without modifying it. Otherwise, first it calls a function \textsc{FindMax}(\(A\)), which scans the array once, finds the index of the maximum element, and returns it. Let this index be \(i_{\text{max}}\). Then \textsc{RSort} swaps \(A[i_{\text{max}}]\) and \(A[n]\) in constant time. Finally it calls \textsc{RSort}(\(A, n - 1\)), and returns \(A\) as returned from the recursive call, without further modification.

Which recurrence relation is applicable to the running time of \textsc{RSort} (for \(n > 1\))? [8 points]

- □ A. \(T(n) = 2T(n - 1) + \Theta(n)\)
- □ B. \(T(n) = 2T(n - 1) + \Theta(1)\)
- ✓ C. \(T(n) = T(n - 1) + \Theta(n)\)
- □ D. \(T(n) = T(n - 1) + \Theta(1)\)
1. Choose all the correct statements. [6 points]

☑ a. There can be no transitions out of a final state in a state-diagram.

☑ b. A state-diagram can have zero or more final states, but exactly one start state.

☐ c. If a state-diagram is deterministic, and it has two states A and B, then there cannot be two different edges between them, both directed from A to B.

☐ d. None of the above.

2. The problem of graph 3-colorability, 3COL, is NP-Complete. Which of the following statements are known to be implied by this? [6 points]

☑ a. If every problem in NP is in P, then 3COL ∈ P.

☑ b. If 3COL ∈ P, then every problem in NP is in P.

☐ c. If some problem in NP is in P, then 3COL ∈ P.

☐ d. None of the above.

3. Consider a recursive sorting algorithm RSort that takes \((A, n)\) where A is an array of size at least \(n\), and sorts the first \(n\) positions of A. (So to sort an entire array using RSort, we can set \(n\) to be the size of the array.)

On input \((A, n)\), RSort works as follows. If \(n = 1\), RSort returns A without modifying it. Otherwise, first it calls a function FINDMAX(A), which scans the array once, finds the index of the maximum element, and returns it. Let this index be \(i_{\text{max}}\). Then RSort swaps \(A[i_{\text{max}}]\) and \(A[n]\) in constant time. Finally it calls RSort\((A, n - 1)\), and returns A as returned from the recursive call, without further modification.

Which recurrence relation is applicable to the running time of RSort (for \(n > 1\))? [8 points]

☐ a. \(T(n) = 2T(n - 1) + \Theta(1)\)

☐ b. \(T(n) = 2T(n - 1) + \Theta(n)\)

☐ c. \(T(n) = T(n - 1) + \Theta(1)\)

☑ d. \(T(n) = T(n - 1) + \Theta(n)\)