Quiz

Score: 11/13

✓ **1. What is the purpose of Apriori analysis in data mining?**

- [✓ To identify the most frequent item set in a transaction database]
- To predict future transactions
- To analyze data in real-time
- To perform sentiment analysis

Explanation

Apriori analysis is used to find frequent item sets in a transaction database. It helps in identifying the most common patterns or relationships between different items.

✓ **2. What does Big O notation represent in algorithm analysis?**

- Best-case time complexity
- [✓ Worst-case time complexity]
- Average-case time complexity
- Space complexity

Explanation

Big O notation is used to describe the upper bound of an algorithm's time complexity. It represents the worst-case scenario for the growth rate of the algorithm's performance.

✓ **3. Which notation represents the lower bound of an algorithm's time complexity?**

- Big O notation
- Theta notation
- [✓ Omega notation]
- Master's theorem

Explanation

Omega notation represents the best-case scenario for the growth rate of an algorithm's performance, i.e., the lower bound of time complexity.

✓

4. **What is the time complexity of an algorithm that performs O(n^2) operations?**

- Logarithmic (O(log n))
- Linear (O(n))
- [✓ Quadratic (O(n^2))]
- Exponential (O(2^n))

Explanation

O(n^2) represents quadratic time complexity, where the number of operations grows quadratically with the input size.

✓

5. **Which method is used to solve recurrence relations by expanding the recursion tree?**

- Apriori analysis
- Substitution method
- [✓ Recursive tree method]
- Master's theorem

Explanation

The recursive tree method involves expanding the tree of recursive calls to analyze the recursive algorithm's time complexity.

✓

6. **What does the Master's theorem provide in the context of algorithm analysis?**

- Optimal solution to all algorithms
- Lower bound of time complexity
- Framework for dynamic programming
- [✓ Method to solve recurrence relations]

Explanation

The Master's theorem provides a straightforward method to solve recurrence relations and analyze the time complexity of divide-and-conquer algorithms.

✗

7. **In the context of algorithm analysis, what does Theta notation represent?**

- Upper bound of time complexity

Explanation

Lower bound of time complexity

Average case time complexity

Tight bound of time complexity ✓

Theta notation represents the tight bound or the average case of an algorithm's time complexity. It provides both upper and lower bound for the growth rate of the algorithm's performance.

✓ **8. Which of the following represents the best-case time complexity of an algorithm?**

Big O notation

Omega notation ✓

Theta notation

Master's theorem

Explanation

The best-case time complexity represents the lower limit of the algorithm's performance and is defined using Omega notation.

✓ **9. What does the Substitution method provide in the context of algorithm analysis?**

Direct calculation of time complexity

Framework for dynamic programming

Method to solve recurrence relations ✓

Analysis of system performance

Explanation

The substitution method is a technique used to solve recurrence relations and analyze the time complexity of recursive algorithms by substituting and solving the recurrence equation.

✓ **10. Which notation provides a tight upper bound for an algorithm's time complexity?**

Big O notation ✓

Omega notation

Theta notation

Explanation

Big O notation provides an upper bound for an algorithm's time complexity, denoting the maximum

Substitution method

growth rate of the algorithm's performance.

✓

11. What is the time complexity of an algorithm that performs O(1) operations?

- Linear (O(n))
- Logarithmic (O(log n))
- [✓ Constant (O(1))]
- Quadratic (O(n^2))

Explanation

O(1) represents constant time complexity, indicating that the number of operations does not depend on the input size.

×

12. What does Theta notation imply about the time complexity of an algorithm?

- Worst-case time complexity
- Best-case time complexity
- Average case time complexity
- [✓ Tight bound of time complexity]

Explanation

Theta notation implies that an algorithm has both an upper and lower bound for its time complexity, indicating a consistent and predictable performance regardless of the input size.

✓

13. Which of the following is true about the Master's theorem?

- It solves all types of recurrence relations
- It provides the exact time complexity for all algorithms
- It is used only for sorting algorithms
- [✓ It analyzes the time complexity of divide-and-conquer algorithms]

Explanation

The Master's theorem provides a concise method to analyze the time complexity of divide-and-conquer algorithms by determining the running time based on the recurrence relation form.