1. How do we handle duplicates in a BST?

A. Insert duplicates on the right subtree.
B. Insert duplicates on the left subtree.
C. Do not allow duplicates.
D. Any of the above, depending on implementation.
Explanation: মডিউলে বলা হইছিলো, **BST** তে ডুপ্লিকেট ভ্যালু নিয়ে সরাসরি কাজ করা হয় না। তবে, ডুপ্লিকেট ভ্যালু ও কাউন্ট ট্রি তে **pair** হিসেবে রাখা যায়। এর উপর ভিত্তি করে কুইজ করা হয়েছে। তবে, ডুপ্লিকেট ভ্যালু **BST-**তে বিভিন্নভাবে হ্যান্ডেল করা যায়**:** ডান সাবট্রিতে, বাম সাবট্রিতে, বা ডুপ্লিকেট এলাউ না করেও।

2. In a BST, which property is true for any node?

A. Left child is always smaller, and right child is always larger.
B. Left child is always larger, and right child is always smaller.
C. All nodes have two children.
D. All nodes are equal.
Explanation: প্রতিটি নোডের বাম সাবট্রিতে ছোট মান এবং ডান সাবট্রিতে বড় মান থাকে। এটাই **BST-**এর মূল নিয়ম।

3. How do you insert a value in a BST?

A. Traverse to the leftmost node and insert the value.
B. Traverse to the rightmost node and insert the value.
C. Compare the value with nodes and insert it in the correct position to maintain BST property.
D. Replace the root node with the new value.
Explanation: BST-এর ডান ও বাম সাবট্রির তুলনা করে উপযুক্ত স্থানে ইনসার্ট করা হয়।

4. What happens to the left subtree during a BST insertion?

A. It remains unchanged.
B. The new value is inserted only if it's smaller than the current node.
C. The new value is inserted only if it's larger than the current node.
D. The entire tree is restructured.
Explanation: BST-তে কোনো মান ইনসার্ট করার সময়, **BST-**এর নিয়ম মেনে চলতে হয়। এই নিয়মে প্রতিটি নোডের বাম সাবট্রিতে কেবলমাত্র ছোট মান ইনসার্ট করা হয় এবং ডান সাবট্রিতে বড় মান ইনসার্ট করা হয়।

5. If the input array is empty, what should be returned when converting to a BST?

A. An empty tree (nullptr Or NULL).
B. A root node with value 0.
C. An exception.
D. A tree with all nodes as nullptr.
Explanation: যদি ইনপুট অ্যারে খালি থাকে, তাহলে কোনো নোড তৈরি করার মতো মান থাকবে না। ফলে, ফাংশনকে একটি **NULL pointer** বা **nullptr** রিটার্ন করতে হবে, যা একটি খালি **tree** কে বোঝায়।

6. In the process of converting a sorted array to a BST, which element is chosen as the root?

A. The smallest element.
B. The largest element.
C. The middle element.
D. Any random element.
Explanation: অ্যারের মাঝখানের উপাদান রুট হিসেবে নির্বাচন করা হয়। এতে অ্যারেটি দুইভাগে বিভক্ত হয়: বাম দিকের অংশটি বাম সাবট্রি হবে এবং ডান দিকের অংশটি ডান সাবট্রি হবে।

7. What is the average-case time complexity of searching in a BST?

A. O(1)
B. O(log n)
C. O(n)
D. O(n^2)
Explanation: BST-তে সার্চ করার গড় টাইম কমপ্লেক্সিটি হল **O(log n),** যেখানে **n** হলো **tree** এর নোড সংখ্যা।

8. What happens if a value greater than the root is inserted in a BST?

A. It becomes the root.
B. It is inserted in the left subtree.
C. It is inserted in the right subtree.
D. It replaces the smallest value in the tree.
Explanation: এটি **BST-**এর মূল বৈশিষ্ট্য, যেখানে প্রতি নোডের বাম সাবট্রিতে ছোট মান এবং ডান সাবট্রিতে বড় মান থাকে। যদি আপনি একটি মান ইনসার্ট করতে চান এবং সেই মান রুটের চেয়ে বড় হয়, তবে সেটি ডান সাবট্রিতে ইনসার্ট করা হবে।

9. When converting a sorted array to a BST, which property of the array is crucial?

A. The array must be sorted in ascending order.
B. The array must have distinct elements.
C. The array must have at least one element.
D. The array must be sorted in descending order.
Explanation: BST তৈরি করার জন্য, অ্যারেটি ঊর্ধ্বমুখীভাবে সজ্জিত **(sorted in ascending order)** থাকা অত্যন্ত গুরুত্বপূর্ণ। কারণ, একটি সজ্জিত অ্যারে থেকে মাঝের উপাদানটি **(middle element)** রুট হিসেবে নির্বাচন করা হয়। যদি অ্যারে সজ্জিত না থাকে, তবে এমন **tree** তৈরি হবে যা ব্যালেন্সড হবে না, এবং সেক্ষেত্রে সার্চ বা অন্যান্য অপারেশন হবে না।

10. If an array of size n is converted into a BST, what will be the time complexity of the conversion process?

A. O(n2)

B. O(n)
C. O(logn)
D. O(nlogn)
Explanation: যখন একটি সজ্জিত অ্যারে থেকে **BST** তৈরি করা হয়**,** তখন রূপান্তর প্রক্রিয়া সাধারণত **O(n)** সময় নিবে। এর কারণ হল**:** অ্যারেটির প্রতিটি উপাদান থেকে একটি নতুন নোড তৈরি করা হয় এবং এটি **BST tree** এর সাথে যুক্ত করা হয়। যদি অ্যারেটিতে **n** সংখ্যক উপাদান থাকে তাহলে **n** সংখ্যক বার নতুন নোড বানাতে হবে এবং তা **BST** তে যুক্ত করতে হবে।