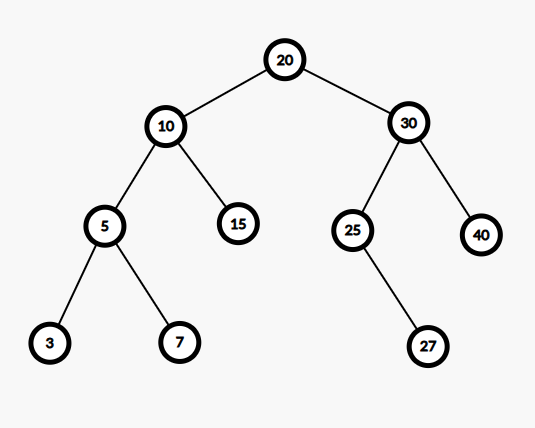
**Question:**

Given the root of the following Binary Search Tree and an integer k, **return the product of all node values that are strictly greater than k.**

**Note:**

* Use **recursion**.
* If **no nodes** are greater than k, return 1.
* The tree will have **distinct positive integers**.
* You may define and use **helper functions**.

**Tree:**



| **Sample Input** | **Sample Output** |
| --- | --- |
| 15 | 20 × 25 × 27 × 30 × 40 = 16200000  **Explanation:** Nodes > 15 |
| 25 | 27 × 30 × 40 = 32400 **Explanation:** Nodes > 25 |
| 35 | 40 = 40  **Explanation:** Only node > 35 |
| 100 | 1  **Explanation:** No nodes > 100 |

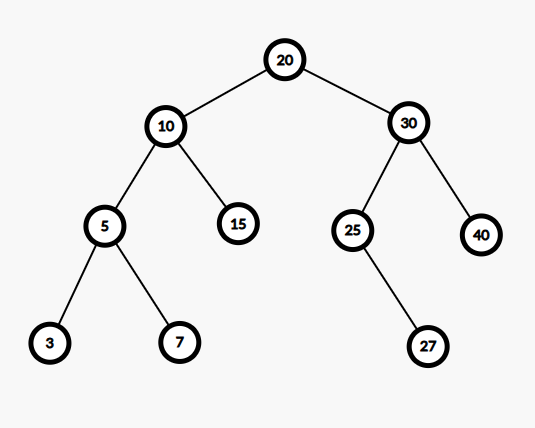
**Question:**

Given the root of the same Binary Search Tree and an integer target, **count the number of nodes whose value is divisible by target.**

**Note:**

* Use **recursion**.
* If **no nodes** are divisible by the target, return 0.
* Assume all values in the tree are **positive integers**.

**Tree:**



| **Sample Input** | **Sample Output** |
| --- | --- |
| 5 | 5  **Explanation:** 5, 10, 15, 20, 25, 30, 40 |
| 3 | 3 **Explanation:** 3, 15, 27 |
| 7 | 1  **Explanation:** Only 7 |
| 13 | 0  **Explanation:** No node is divisible by 13 |