Problem 2271: Maximum White Tiles Covered by a Carpet

Problem Information

Difficulty: Medium
Acceptance Rate: 35.43%
Paid Only: No
Tags: Array, Binary Search, Greedy, Sliding Window, Sorting, Prefix Sum

Problem Description

You are given a 2D integer array `tiles` where `tiles[i] = [li, ri]` represents that every tile `j` in the range `li <= j <= ri` is colored white.

You are also given an integer `carpetLen`, the length of a single carpet that can be placed **anywhere**.

Return _the**maximum** number of white tiles that can be covered by the carpet_.

Example 1:

Input: tiles = [[1,5],[10,11],[12,18],[20,25],[30,32]], carpetLen = 10 **Output:** 9 **Explanation:** Place the carpet starting on tile 10. It covers 9 white tiles, so we return 9. Note that there may be other places where the carpet covers 9 white tiles. It can be shown that the carpet cannot cover more than 9 white tiles.

Example 2:

Input: tiles = [[10,11],[1,1]], carpetLen = 2 **Output:** 2 **Explanation:** Place the carpet starting on tile 10. It covers 2 white tiles, so we return 2.

Constraints:

* `1 <= tiles.length <= 5 * 104` * `tiles[i].length == 2` * `1 <= li <= ri <= 109` * `1 <= carpetLen <= 109` * The `tiles` are **non-overlapping**.

Code Snippets

C++:

```
class Solution {
public:
int maximumWhiteTiles(vector<vector<int>>& tiles, int carpetLen) {

}
};
```

Java:

```
class Solution {
public int maximumWhiteTiles(int[][] tiles, int carpetLen) {

}
}
```

Python3:

```
class Solution:
def maximumWhiteTiles(self, tiles: List[List[int]], carpetLen: int) -> int:
```