Problem 2188: Minimum Time to Finish the Race

Problem Information

Difficulty: Hard
Acceptance Rate: 42.84%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given a **0-indexed** 2D integer array `tires` where `tires[i] = [fi, ri]` indicates that the `ith` tire can finish its `xth` successive lap in `fi * ri(x-1)` seconds.

* For example, if `fi = 3` and `ri = 2`, then the tire would finish its `1st` lap in `3` seconds, its `2nd` lap in `3 * 2 = 6` seconds, its `3rd` lap in `3 * 22 = 12` seconds, etc.

You are also given an integer `changeTime` and an integer `numLaps`.

The race consists of `numLaps` laps and you may start the race with **any** tire. You have an **unlimited** supply of each tire and after every lap, you may **change** to any given tire (including the current tire type) if you wait `changeTime` seconds.

Return _the**minimum** time to finish the race._

Example 1:

Input: tires = [[2,3],[3,4]], changeTime = 5, numLaps = 4 **Output:** 21 **Explanation:** Lap 1: Start with tire 0 and finish the lap in 2 seconds. Lap 2: Continue with tire 0 and finish the lap in 2 * 3 = 6 seconds. Lap 3: Change tires to a new tire 0 for 5 seconds and then finish the lap in another 2 seconds. Lap 4: Continue with tire 0 and finish the lap in 2 * 3 = 6 seconds. Total time = 2 + 6 + 5 + 2 + 6 = 21 seconds. The minimum time to complete the race is 21 seconds.

Example 2:

Input: tires = [[1,10],[2,2],[3,4]], changeTime = 6, numLaps = 5 **Output:** 25 **Explanation:** Lap 1: Start with tire 1 and finish the lap in 2 seconds. Lap 2: Continue with tire 1 and finish the lap in 2 * 2 = 4 seconds. Lap 3: Change tires to a new tire 1 for 6 seconds and then finish the lap in another 2 seconds. Lap 4: Continue with tire 1 and finish the lap in 2 * 2 = 4 seconds. Lap 5: Change tires to tire 0 for 6 seconds then finish the lap in another 1 second. Total time = 2 + 4 + 6 + 2 + 4 + 6 + 1 = 25 seconds. The minimum time to complete the race is 25 seconds.

Constraints:

* `1 <= tires.length <= 105` * `tires[i].length == 2` * `1 <= fi, changeTime <= 105` * `2 <= ri <= 105` * `1 <= numLaps <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int minimumFinishTime(vector<vector<int>>& tires, int changeTime, int numLaps) {

}
};
```

Java:

```
class Solution {
public int minimumFinishTime(int[][] tires, int changeTime, int numLaps) {

}
}
```

Python3:

```
class Solution:
def minimumFinishTime(self, tires: List[List[int]], changeTime: int, numLaps: int) -> int:
```