6011. Minimum Time to Finish the Race

My Submissions (/contest/weekly-contest-282/problems/minimum-time-to-finish-the-race/submissions/)

Back to Contest (/contest/weekly-contest-282/)

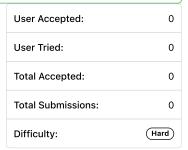
You are given a **0-indexed** 2D integer array tires where tires[i] = [f_i , r_i] indicates that the ith tire can finish its x^{th} successive lap in $f_i * r_i^{(x-1)}$ seconds.

• For example, if $f_i = 3$ and $r_i = 2$, then the tire would finish its 1^{st} lap in 3 seconds, its 2^{nd} lap in 3 * 2 = 6 seconds, its 3^{rd} lap in $3 * 2^2 = 12$ 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 <= 10<sup>5</sup>
    tires[i].length == 2
    1 <= f<sub>i</sub>, changeTime <= 10<sup>5</sup>
    2 <= r<sub>i</sub> <= 10<sup>5</sup>
```

```
• 1 <= numLaps <= 1000
```

```
C
JavaScript
   const MAX = Number.MAX_SAFE_INTEGER;
2 •
    const minimumFinishTime = (tires, changeTime, numLaps) => {
3
        let min = Array(20).fill(MAX), dp = Array(numLaps + 1).fill(MAX);
 4
        min[0] = 0;
5 •
        for (const [f, r] of tires) {
6
            let sum = 0;
 7
            for (let i = 1; i < 20; i++) {
                sum += f * r ** (i - 1);
8
9
                min[i] = Math.min(min[i], sum);
10
                if (sum > 1e6) break;
11
            }
12
        }
```

```
dp[0] = 0;
for (let i = 1; i <= numLaps; i++) {
    for (let j = 1; j < 20; j++) {
        if (i >= j) dp[i] = Math.min(dp[i], dp[i - j] + min[j] + changeTime);
}

return dp[numLaps] - changeTime;
};
```

☐ Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/649684468/) •

More Details > (/submissions/detail/649684468/)

Share your acceptance!

∢1

Copyright © 2022 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)