2013. Minimum Skips to Arrive at Meeting On Time

Difficulty : Hard

https://leetcode.com/problems/minimum-skips-to-arrive-at-meeting-on-time

You are given an integer `hoursBefore`, the number of hours you have to travel to your meeting. To arrive at your meeting, you have to travel through `n` roads. The road lengths are given as an integer array `dist` of length `n`, where `dist[i]` describes the length of the ith road in **kilometers**. In addition, you are given an integer `speed`, which is the speed (in **km/h**) you will travel at.

After you travel road `i`, you must rest and wait for the **next integer hour** before you can begin traveling on the next road. Note that you do not have to rest after traveling the last road because you are already at the meeting.

- For example, if traveling a road takes `1.4` hours, you must wait until the `2` hour mark before traveling the next road. If traveling a road takes exactly `2` hours, you do not need to wait.

However, you are allowed to **skip** some rests to be able to arrive on time, meaning you do not need to wait for the next integer hour. Note that this means you may finish traveling future roads at different hour marks.

- For example, suppose traveling the first road takes `1.4` hours and traveling the second road takes `0.6` hours. Skipping the rest after the first road will mean you finish traveling the second road right at the `2` hour mark, letting you start traveling the third road immediately.

Return *the **minimum number of skips required** to arrive at the meeting on time, or* `-1` *if it is **impossible***.

Example 1:

```
Input: dist = [1,3,2], speed = 4, hoursBefore = 2
Output: 1
Explanation:
Without skipping any rests, you will arrive in (1/4 + 3/4) + (3/4 + 1/4) + (2/4) = 2.5 hours.
You can skip the first rest to arrive in ((1/4 + 0) + (3/4 + 0)) + (2/4) = 1.5 hours.
Note that the second rest is shortened because you finish traveling the second road at an integer hour due to skipping the first rest.
```

Example 2:

```
Input: dist = [7,3,5,5], speed = 2, hoursBefore = 10
Output: 2
Explanation:
Without skipping any rests, you will arrive in (7/2 + 1/2) + (3/2 + 1/2) + (5/2 + 1/2) + (5/2) = 11.5 hours.
You can skip the first and third rest to arrive in ((7/2 + 0) + (3/2 + 0)) + ((5/2 + 0) + (5/2)) = 10 hours.
```

Example 3:

```
Input: dist = [7,3,5,5], speed = 1, hoursBefore = 10
Output: -1
Explanation: It is impossible to arrive at the meeting on time even if you skip all the rests.
```

Constraints:

- `n == dist.length`
- `1 <= n <= 1000`
- `1 <= dist[i] <= 10^5`
- `1 <= speed <= 10^6`
- `1 <= hoursBefore <= 10^7`