1024. Video Stitching

Description

You are given a series of video clips from a sporting event that lasted `time` seconds. These video clips can be overlapping with each other and have varying lengths.

Each video clip is described by an array `clips` where `clips[i] = [start_i, end_i]` indicates that the ith clip started at `start_i` and ended at `end_i`.

We can cut these clips into segments freely.

- For example, a clip `[0, 7]` can be cut into segments `[0, 1] + [1, 3] + [3, 7]`.

Return *the minimum number of clips needed so that we can cut the clips into segments that cover the entire sporting event* `[0, time]`. If the task is impossible, return `-1`.

Example 1:

```
Input: clips = [[0,2],[4,6],[8,10],[1,9],[1,5],[5,9]], time = 10
Output: 3
Explanation: We take the clips [0,2], [8,10], [1,9]; a total of 3 clips.
Then, we can reconstruct the sporting event as follows:
We cut [1,9] into segments [1,2] + [2,8] + [8,9].
Now we have segments [0,2] + [2,8] + [8,10] which cover the sporting event [0, 10].
```

Example 2:

```
Input: clips = [[0,1],[1,2]], time = 5
Output: -1
Explanation: We cannot cover [0,5] with only [0,1] and [1,2].
```

Example 3:

```
Input: clips = [[0,1],[6,8],[0,2],[5,6],[0,4],[0,3],[6,7],[1,3],[4,7],[1,4],[2,5],[2,6],[3,4],[4,5],[5,7],[6,9]], time = 9
Output: 3
Explanation: We can take clips [0,4], [4,7], and [6,9].
```

Constraints:

- `1 <= clips.length <= 100`
- `0 <= start_i <= end_i <= 100`
- `1 <= time <= 100`