2254. Design Video Sharing Platform

Premium

Hard · Topics · Companies · Hint

You have a video sharing platform where users can upload and delete videos. Each `video` is a **string** of digits, where the `i`th digit of the string represents the content of the video at minute `i`. For example, the first digit represents the content at minute `0` in the video, the second digit represents the content at minute `1` in the video, and so on. Viewers of videos can also like and dislike videos. Internally, the platform keeps track of the **number of views, likes, and dislikes** on each video.

When a video is uploaded, it is associated with the smallest available integer `videoId` starting from `0`. Once a video is deleted, the `videoId` associated with that video can be reused for another video.

Implement the `VideoSharingPlatform` class:

- `VideoSharingPlatform()` Initializes the object.
- `int upload(String video)` The user uploads a `video`. Return the `videoId` associated with the video.
- `void remove(int videoId)` If there is a video associated with `videoId`, remove the video.
- `String watch(int videoId, int startMinute, int endMinute)` If there is a video associated with `videoId`, increase the number of views on the video by `1` and return the substring of the video string starting at `startMinute` and ending at `min(endMinute, video.length - 1)` (**inclusive**). Otherwise, return `"-1"`.
- `void like(int videoId)` Increases the number of likes on the video associated with `videoId` by `1` if there is a video associated with `videoId`.
- `void dislike(int videoId)` Increases the number of dislikes on the video associated with `videoId` by `1` if there is a video associated with `videoId`.
- `int[] getLikesAndDislikes(int videoId)` Return a **0-indexed** integer array `values` of length `2` where `values[0]` is the number of likes and `values[1]` is the number of dislikes on the video associated with `videoId`. If there is no video associated with `videoId`, return `[-1]`.
- `int getViews(int videoId)` Return the number of views on the video associated with `videoId`, if there is no video associated with `videoId`, return `-1`.

Example 1:

```
Input
["VideoSharingPlatform", "upload", "upload", "remove", "remove", "upload", "watch", "watch", "like", "dislike", "dislike", "getLikesAndDislikes", "getViews"]
[[], ["123"], ["456"], [4], [0], ["789"], [1, 0, 5], [1, 0, 1], [1], [1], [1], [1], [1]]
Output
[null, 0, 1, null, null, 0, "456", "45", null, null, null, [1, 2], 2]

Explanation
VideoSharingPlatform videoSharingPlatform = new VideoSharingPlatform();
videoSharingPlatform.upload("123");          // The smallest available videoId is 0, so return 0.
videoSharingPlatform.upload("456");          // The smallest available videoId is 1, so return 1.
videoSharingPlatform.remove(4);              // There is no video associated with videoId 4, so do nothing.
videoSharingPlatform.remove(0);              // Remove the video associated with videoId 0.
videoSharingPlatform.upload("789");          // Since the video associated with videoId 0 was deleted,
                                             // 0 is the smallest available videoId, so return 0.
videoSharingPlatform.watch(1, 0, 5);         // The video associated with videoId 1 is "456".
                                             // The video from minute 0 to min(5, 3 - 1) = 2 is "456", so return "456".
videoSharingPlatform.watch(1, 0, 1);         // The video associated with videoId 1 is "456".
                                             // The video from minute 0 to min(1, 3 - 1) = 1 is "45", so return "45".
videoSharingPlatform.like(1);                // Increase the number of likes on the video associated with videoId 1.
videoSharingPlatform.dislike(1);             // Increase the number of dislikes on the video associated with videoId 1.
videoSharingPlatform.dislike(1);             // Increase the number of dislikes on the video associated with videoId 1.
videoSharingPlatform.getLikesAndDislikes(1); // There is 1 like and 2 dislikes on the video associated with videoId 1, so return [1, 2].
videoSharingPlatform.getViews(1);            // The video associated with videoId 1 has 2 views, so return 2.
```

Example 2:

```
Input
["VideoSharingPlatform", "remove", "watch", "like", "dislike", "getLikesAndDislikes", "getViews"]
[[], [0], [0, 0, 1], [0], [0], [0], [0]]
Output
[null, null, "-1", null, null, [-1], -1]

Explanation
VideoSharingPlatform videoSharingPlatform = new VideoSharingPlatform();
videoSharingPlatform.remove(0);              // There is no video associated with videoId 0, so do nothing.
videoSharingPlatform.watch(0, 0, 1);         // There is no video associated with videoId 0, so return "-1".
videoSharingPlatform.like(0);                // There is no video associated with videoId 0, so do nothing.
videoSharingPlatform.dislike(0);             // There is no video associated with videoId 0, so do nothing.
videoSharingPlatform.getLikesAndDislikes(0); // There is no video associated with videoId 0, so return [-1].
videoSharingPlatform.getViews(0);            // There is no video associated with videoId 0, so return -1.
```

Constraints:

- $1 <= \texttt{video.length} <= 10^5$
- The sum of `video.length` over all calls to `upload` does not exceed 10^5
- `video` consists of digits.
- $0 <= \texttt{videoId} <= 10^5$
- $0 <= \texttt{startMinute} < \texttt{endMinute} < 10^5$
- `startMinute < video.length`
- The sum of `endMinute - startMinute` over all calls to `watch` does not exceed 10^5.
- At most 10^5 calls **in total** will be made to all functions.

Seen this question in a real interview before? 1/5

Accepted 3.9K | Submissions 6.2K | Acceptance Rate 63.0%

Topics

Companies

0 - 6 months

Hint 1

How can we keep track of the videoId's of removed videos?

Hint 2

When you remove a video, keep track of its videoId in an array called usedId. When a new video is uploaded, first check if usedId contains any elements, if it does, use the smallest value in usedId.

Similar Questions

Tweet Counts Per Frequency	Medium
Design Browser History	Medium
Most Popular Video Creator	Medium

Discussion (8)