Problem 2254: Design Video Sharing Platform

Problem Information

Difficulty: Hard
Acceptance Rate: 63.74%
Paid Only: Yes
Tags: Hash Table, Stack, Design, Ordered Set

Problem Description

You have a video sharing platform where users can upload and delete videos. Each `video` is a **string** of digits, where the `ith` 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 <= video.length <= 105` * The sum of `video.length` over all calls to `upload` does not exceed `105` * `video` consists of digits. * `0 <= videoId <= 105` * `0 <= startMinute < endMinute < 105` * `startMinute < video.length` * The sum of `endMinute - startMinute` over all calls to `watch` does not exceed `105`. * At most `105` calls **in total** will be made to all functions.

Code Snippets

C++:

```
class VideoSharingPlatform {
public:
VideoSharingPlatform() {

}

int upload(string video) {

}

void remove(int videoId) {

}

string watch(int videoId, int startMinute, int endMinute) {

}

void like(int videoId) {

}

void dislike(int videoId) {

}

vector<int> getLikesAndDislikes(int videoId) {

}

int getViews(int videoId) {
```

```
}
};

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform* obj = new VideoSharingPlatform();
* int param_1 = obj->upload(video);
* obj->remove(videoId);
* string param_3 = obj->watch(videoId,startMinute,endMinute);
* obj->like(videoId);
* obj->dislike(videoId);
* vector<int> param_6 = obj->getLikesAndDislikes(videoId);
* int param_7 = obj->getViews(videoId);
*/
```

Java:

```
class VideoSharingPlatform {

public VideoSharingPlatform() {

}

public int upload(String video) {

}

public void remove(int videoId) {

}

public String watch(int videoId, int startMinute, int endMinute) {

}

public void like(int videoId) {

}

public void dislike(int videoId) {
```

```
}

public int[] getLikesAndDislikes(int videoId) {

}

public int getViews(int videoId) {

}
}

/**
* Your VideoSharingPlatform object will be instantiated and called as such:
* VideoSharingPlatform obj = new VideoSharingPlatform();
* int param_1 = obj.upload(video);
* obj.remove(videoId);
* String param_3 = obj.watch(videoId,startMinute,endMinute);
* obj.like(videoId);
* obj.dislike(videoId);
* int[] param_6 = obj.getLikesAndDislikes(videoId);
* int param_7 = obj.getViews(videoId);
*/
```

Python3:

```
class VideoSharingPlatform:

def __init__(self):


def upload(self, video: str) -> int:


def remove(self, videoId: int) -> None:


def watch(self, videoId: int, startMinute: int, endMinute: int) -> str:


def like(self, videoId: int) -> None:
```

```
def dislike(self, videoId: int) -> None:


def getLikesAndDislikes(self, videoId: int) -> List[int]:


def getViews(self, videoId: int) -> int:



# Your VideoSharingPlatform object will be instantiated and called as such:
# obj = VideoSharingPlatform()
# param_1 = obj.upload(video)
# obj.remove(videoId)
# param_3 = obj.watch(videoId,startMinute,endMinute)
# obj.like(videoId)
# obj.dislike(videoId)
# param_6 = obj.getLikesAndDislikes(videoId)
# param_7 = obj.getViews(videoId)
```