3408. Design Task Manager

Solved ●

Medium Topics Companies

There is a task management system that allows users to manage their tasks, each associated with a priority. The system should efficiently handle adding, modifying, executing, and removing tasks.

Implement the `TaskManager` class:

- `TaskManager(vector<vector<int>>& tasks)` initializes the task manager with a list of user-task-priority triples. Each element in the input list is of the form `[userId, taskId, priority]`, which adds a task to the specified user with the given priority.
- `void add(int userId, int taskId, int priority)` adds a task with the specified `taskId` and `priority` to the user with `userId`. It is **guaranteed** that `taskId` does not *exist* in the system.
- `void edit(int taskId, int newPriority)` updates the priority of the existing `taskId` to `newPriority`. It is **guaranteed** that `taskId` *exists* in the system.
- `void rmv(int taskId)` removes the task identified by `taskId` from the system. It is **guaranteed** that `taskId` *exists* in the system.
- `int execTop()` executes the task with the **highest** priority across all users. If there are multiple tasks with the same **highest** priority, execute the one with the highest `taskId`. After executing, the `taskId` is **removed** from the system. Return the `userId` associated with the executed task. If no tasks are available, return -1.

Note that a user may be assigned multiple tasks.

Example 1:

> **Input:**
> ["TaskManager", "add", "edit", "execTop", "rmv", "add", "execTop"]
> [[[[1, 101, 10], [2, 102, 20], [3, 103, 15]]], [4, 104, 5], [102, 8], [], [101], [5, 105, 15], []]
>
> **Output:**
> [null, null, null, 3, null, null, 5]
>
> **Explanation**
>
> TaskManager taskManager = new TaskManager([[1, 101, 10], [2, 102, 20], [3, 103, 15]]); // Initializes with three tasks for Users 1, 2, and 3.
> taskManager.add(4, 104, 5); // Adds task 104 with priority 5 for User 4.
> taskManager.edit(102, 8); // Updates priority of task 102 to 8.
> taskManager.execTop(); // return 3. Executes task 103 for User 3.
> taskManager.rmv(101); // Removes task 101 from the system.
> taskManager.add(5, 105, 15); // Adds task 105 with priority 15 for User 5.
> taskManager.execTop(); // return 5. Executes task 105 for User 5.

Constraints:

- $1 <= tasks.length <= 10^5$
- $0 <= userId <= 10^5$
- $0 <= taskId <= 10^5$
- $0 <= priority <= 10^9$
- $0 <= newPriority <= 10^9$
- At most $2 * 10^5$ calls will be made in **total** to `add`, `edit`, `rmv`, and `execTop` methods.
- The input is generated such that `taskId` will be valid.

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

Yes No

Accepted 12.9K Submissions 38K Acceptance Rate 34.1%

Topics

Companies

Discussion (35)