Problem Definition:

設計一資料結構可以用作存 time-based key-value 資料，其中，每个 key 可存多个 values 在不同的 time stamp 下，並依 key 和 time stamp 來拿到对應 value.

① Data: key, value, timestamp

② Operation: (1). set (key, value, timestamp)

將 value 存入对應的 key 上且在 timestamp 時間

(2). get (key, timestamp)

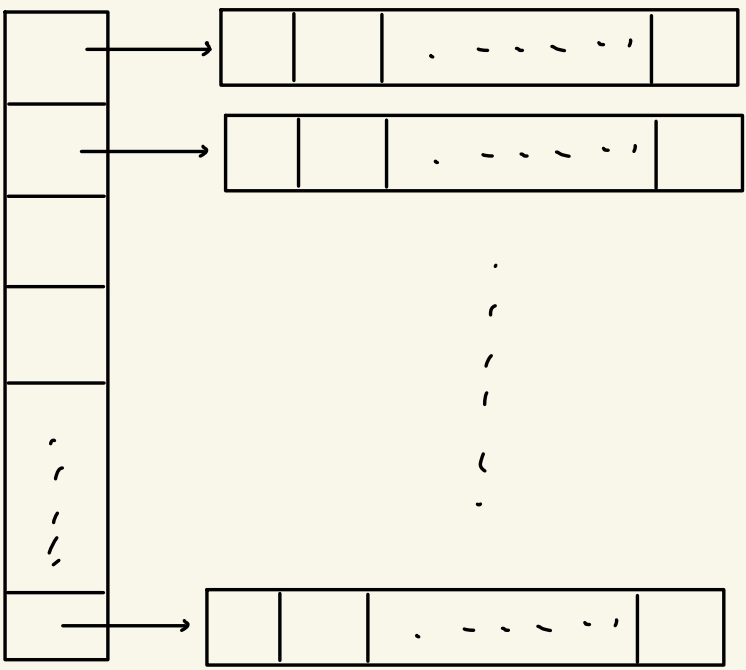
取得在 key 中有的 value 中

$time_i \le$ timestamp 最大的那个.

若無，則回傳空字串.

如何設計資料結構?

① key - value 資料 ⇒ hashing



② 用 hash(key) 取得对應 value 在資料指標.

⇒ Time: O(1)

Question: 若有 conflict 怎麼辦?

② 取得对应 key 在陣列後，在此用 timestamp 做 binary search.

∀ Time : O(lg m)

如此一來，為了维持陣列一直保持排序，插入時，也要找到小於等於目前 time stamp 最大的位置.

Binary Search 設計：

以往查找 or 插入 time stamp 為 target.

a. nums[mid] ≤ target ∃ [mid, right]

b. nums[mid] > target ∃ [left, mid-1]

I. Set Operation:

① 取 pointer to array = O(1) in amortized cost

② 找到插入 index = O(log(L)), L: 此 array 長度

③ 插入 = O(L)

= O(L log L)

II. Get Operation:

① 取 pointer to array = O(1)

② 找到位置 = O(log(L))

> O(log(L))

Implementation Details:

① Hash Map 中應儲 pointer to array

而非 array 自身

② 包个 pointed array 要儲: 1. value: string

2. time stamp

⑩. 已知所有 set 的 time stamp 皆 strictly increasing.

∴ set operation 中可以不用做 binary search.

⇒ set 時間可減少!