5694. Design Authentication Manager

My Submissions (/contest/biweekly-contest-48/problems/design-authentication-manager/submissions/) | Back to Contest (/contest/biweekly-contest-48/)
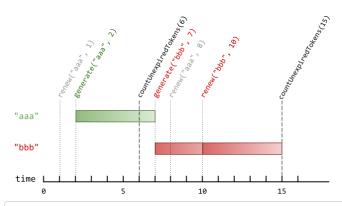
There is an authentication system that works with authentication tokens. For each session, the user will receive a new authentication token that will expire `timeToLive` seconds after the `currentTime`. If the token is renewed, the expiry time will be **extended** to expire `timeToLive` seconds after the (potentially different) `currentTime`.

Implement the `AuthenticationManager` class:

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

- `AuthenticationManager(int timeToLive)` constructs the `AuthenticationManager` and sets the `timeToLive`.
- `generate(string tokenId, int currentTime)` generates a new token with the given `tokenId` at the given `currentTime` in seconds.
- `renew(string tokenId, int currentTime)` renews the **unexpired** token with the given `tokenId` at the given `currentTime` in seconds. If there are no unexpired tokens with the given `tokenId`, the request is ignored, and nothing happens.
- `countUnexpiredTokens(int currentTime)` returns the number of **unexpired** tokens at the given currentTime.

Note that if a token expires at time `t`, and another action happens on time `t` (`renew` or `countUnexpiredTokens`), the expiration takes place **before** the other actions.

Example 1:


```
Input
["AuthenticationManager", "renew", "generate", "countUnexpiredTokens", "generate", "renew", "renew", "countUnexpiredTokens"]
[[5], ["aaa", 1], ["aaa", 2], [6], ["bbb", 7], ["aaa", 8], ["bbb", 10], [15]]
Output
[null, null, null, 1, null, null, null, 0]

Explanation
AuthenticationManager authenticationManager = new AuthenticationManager(5); // Constructs the AuthenticationManager with time
ToLive = 5 seconds.
authenticationManager.renew("aaa", 1); // No token exists with tokenId "aaa" at time 1, so nothing happens.
authenticationManager.generate("aaa", 2); // Generates a new token with tokenId "aaa" at time 2.
authenticationManager.countUnexpiredTokens(6); // The token with tokenId "aaa" is the only unexpired one at time 6, so return 1
authenticationManager.generate("bbb", 7); // Generates a new token with tokenId "bbb" at time 7.
authenticationManager.renew("aaa", 8); // The token with tokenId "aaa" expired at time 7, and 8 >= 7, so at time 8 the renew re
authenticationManager.renew("bbb", 10); // The token with tokenId "bbb" is unexpired at time 10, so the renew request is fulfil
authenticationManager.countUnexpiredTokens(15); // The token with tokenId "bbb" expires at time 15, and the token with tokenId
```

Constraints:

- $1 <= timeToLive <= 10^8$
- $1 <= currentTime <= 10^8$
- $1 <= tokenId.length <= 5$
- `tokenId` consists only of lowercase letters.
- All calls to `generate` will contain unique values of `tokenId`.
- The values of `currentTime` across all the function calls will be **strictly increasing**.
- At most `2000` calls will be made to all functions combined.

JavaScript ▾

```javascript
 1 ▾ function AuthenticationManager(timeToLive) {
 2       this.m = new Map();
 3       this.last = timeToLive;
 4   };
 5
 6 ▾ AuthenticationManager.prototype.generate = function (tokenId, currentTime) {
 7       this.m.set(tokenId, currentTime);
 8   };
 9
10 ▾ AuthenticationManager.prototype.renew = function (tokenId, currentTime) {
11       if (!this.m.has(tokenId)) return;
12       let expire = this.m.get(tokenId) + this.last;
13       if (expire > currentTime) this.m.set(tokenId, currentTime);
14   };
15
16 ▾ AuthenticationManager.prototype.countUnexpiredTokens = function (currentTime) {
17       let cnt = 0;
18 ▾     for (const [k, v] of this.m) {
19           let expire = v + this.last;
20           // pr(expire, currentTime);
21           if (expire > currentTime) cnt++;
22       }
23       return cnt;
24   };
```

☐ **Custom Testcase**   (Use Example Testcases)

● Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/470191498/) ❓   (More Details ❯ (/submissions/detail/470191498/))

Share your acceptance!

◀ 1