

yoj814 History Report

2021201709 李俊霖

算法说明

基本思路：

- 使用对大素数取模的方式，建立从学生id到哈希表的映射（哈希函数）；
- 解决哈希冲突的方法：
 - 链地址法：
 - 插入：先根据查找链找到最末尾的节点。如果要插入的节点未冲突，直接插入即可。如果待插入节点冲突，将now接到pre后面，形成查找链。
 - 查找：根据查找链在哈希表中找到符合id的值。
 - 线性试探法：
 - 插入：调用 `probe4free(key)` 函数，为新词条找到空桶，然后插入新元素。 `probe4free(key)` 函数中，如果该桶内已经有存在值，跳过，桶的下标加1，依次往后继续找空桶。
 - 查找：调用 `probe4hit(key)` 函数，沿着关键码key的查找链顺序查找。 `probe4hit(key)` 函数中，当桶不为空且里面存的关键码不是我们要寻找的关键码时，桶的下标加1，依次往后继续寻找。
 - 双向平方试探法：
 - 查找和插入的基本算法和线性试探法一致；不同的是，对于两个命中函数，如果该桶内已经有存在值，跳过，桶的下标以加/减递增值的平方往两边试探桶单元。当桶不为空且里面存的关键码不是我们要寻找的关键码时，桶的下标桶的下标以加/减递增值的平方往两边试探桶单元，依次往后继续寻找。
- 时间更新：对相同的id进行查询时，更新最后一次出现的时间。

对于以上三种冲突解决方法，笔者在yoj上各提交了一个版本的AC代码。

复杂度分析

- 插入：在理想随机的情况下，每次插入的渐进时间复杂度为 $O(1)$ 。
- 查询：每次查询的渐进时间复杂度为 $O(1)$ 。
- 更新：在哈希表中找到该元素，并更新最后一次查找的时间，渐进时间复杂度为 $O(1)$ 。