

## 第 3-3 讲: 均摊分析

姓名: 林凡琪 学号: 211240042

评分: \_\_\_\_\_ 评阅: \_\_\_\_\_

2022 年 9 月 21 日

请独立完成作业, 不得抄袭。  
若得到他人帮助, 请致谢。  
若参考了其它资料, 请给出引用。  
鼓励讨论, 但需独立书写解题过程。

# 1 作业 (必做部分)

### 题目 1 (TC 17.1-3)

Suppose we perform a sequence of  $n$  operations on a data structure in which the  $i$ th operation costs  $i$  if  $i$  is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

解答:

$$\begin{aligned} S &\leq \sum_{i=0}^{\log_2 n} \frac{n}{2^i} + n - \lfloor \log_2 n \rfloor \\ &\leq 3n \\ &= O(n) \end{aligned}$$

单个操作复杂度  $\frac{O(n)}{n} = O(1)$

---

### 题目 2 (TC 17.2-2)

Redo Exercise 17.1-3 using an accounting method of analysis.

解答:

设平摊代价为:

对于 2 的幂次方 (不含 1): 0

对于其他数: 支付 1, 留下余额 4

对于数字 2, 数字 1 的余额足以将其支付。

对于数字  $2^n (n > 1)$ ,  $(2^{n-1}, 2^n)$  之间的整数足以将其支付。(当  $n \geq 2$  时,  $4 * (2^n - 2^{n-1} - 1) \geq 2^n$ )

故总代价  $S \leq 5n$

单个操作复杂度  $\frac{O(n)}{n} = O(1)$

---

**题目 3 (TC 17.4-1)**

Suppose that we wish to implement a dynamic, open-address hash table. Why might we consider the table to be full when its load factor reaches some value that is strictly less than 1? Describe briefly how to make insertion into a dynamic, open-address hash table run in such a way that the expected value of the amortized cost per insertion is  $O(1)$ . Why is the expected value of the actual cost per insertion not necessarily  $O(1)$  for all insertions?

**解答:**

当  $\alpha$  接近 1 时, 平均次数  $\frac{1}{1-\alpha}$  将会非常大, 从而运行效率下降。

在动态开放寻址中, 取扩张的  $\alpha$  为  $\frac{1}{2}$ , 收缩的  $\alpha$  为  $\frac{1}{4}$  即可。

在大小为  $n$ , 有  $\frac{n}{2} - 1$  个元素的表中进行插入时, 单次插入可能达到  $O(n)$ 。在算法的实际分析中, 只需要代价均摊  $O(1)$  即可。

## 2 作业 (选做部分)

### 3 Open Topics

**Open Topics 1 (Binomial heap)**

请介绍 Binomial heap 的结构以及主要操作, 分析其时间复杂度。

参考资料

- [https://en.wikipedia.org/wiki/Binomial\\_heap](https://en.wikipedia.org/wiki/Binomial_heap)

**Open Topics 2 (Day-Stout-Warren algorithm)**

请介绍 Day-Stout-Warren algorithm, 分析其时间复杂度。

参考资料

- [https://en.wikipedia.org/wiki/Day-Warren\\_algorithm](https://en.wikipedia.org/wiki/Day-Warren_algorithm)

## 4 反馈