算法设计与分析第七章作业

苏亦凡　　计科12班　　200111229

1

假设我们对一个数据结构执行 n 次操作，如果 i 是 2 的乘方则第 i 个操作的开销为 i ，否则为 1 。分别使用聚集法、会计法和势能法分析操作的平摊代价。

聚集法

设第 i 个操作的实际代价为 c_i ，有：

$$c_i = \begin{cases} i, n\text{为2的幂} \\ 1, \text{其他} \end{cases}$$

i 次操作的总代价为

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i + \sum_{i=1}^{n-\lfloor \log_2 n \rfloor} 1 \leq 2n - 1 + n - \lfloor \log_2 n \rfloor = O(n)$$

故平摊代价为 $O(1)$

会计法

赋予每次操作平摊代价 3

当 $i \neq 2^k$ 时，有 $3 - 1 \geq 0$ ，在这些操作上存款只会增加。

当 $i = 2^k$ 时：

1. $k = 0$ 显然有存款。
2. 设 $i = 2^k$ 时有存款，则 $i = 2^{k+1}$ 时有：

$$W \geq 0 + 3 \cdot 2^k - (2^k - 1) - 2^{k+1} = 1 \geq 0$$

综合1，2，当 $i = 2^k$ 时有存款。

综上，任何操作后存款总额均不会出现负数。

总平摊代价为 $3n$ ，平摊代价为 $O(1)$

势能法

定义：

$\Phi(D_0) = 0$

$\Phi(D_i) = 2i - 2 \cdot 2^{\lfloor \log_2 i \rfloor} \geq 0$

- 若 $i \neq 2^k$,$c_i = 1 + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 2 = 3$
- 若 $i = 2^k$，$c_i = i + \Phi(D_i) - \Phi(D_{i-1}) = i + (2i - 2 \cdot 2^k) - (2(i-1) - 2 \cdot 2^{k-1}) = 2$

因此平摊代价为 $O(1)$

2

Bill 提出了一种称为翻转栈的数据结构，该结构仅支持 `Flip_push()` 操作。 每次执行 `Flip_push()` 时，首先入栈，然后检查栈中的对象数是否为 2 的幂。 如果是，则将翻转栈中的所有对象。

例如，我们使用 `Flip_push()` 将对象 1、2、3 和 4 压入栈。堆栈的内容变化（从下至上）如下：

$(1) \Rightarrow (2,1) \Rightarrow (2,1,3) \Rightarrow (4,3,1,2)$

你需要使用分别使用聚集法、会计法和势能法分析 `Flipping_push()` 函数的摊销成本。

堆栈反转的成本等于堆栈中现有对象的数量。

聚集法

此情景中栈元素只进不出，于是有实际代价

$$c_i = \begin{cases} i+1 & ,i\text{为2的幂} \\ 1 & ,\text{其他} \end{cases}$$

i 次操作的总代价为

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i + \sum_{i=1}^{n} 1 \leq 2 \cdot 2^{\lfloor \log_2 n \rfloor} - 1 + n \leq 3n - 1 = O(n)$$

会计法

赋予每次操作平摊代价 3

当 $i \neq 2^k$ 时，有 $3-1 \geq 0$ ，在这些操作上存款只会增加。

当 $i = 2^k$ 时:

1. $k=0$ 显然有存款。
2. 设 $i = 2^k$ 时有存款，则 $i = 2^{k+1}$ 时有:

$$W \geq 0 + 3 \cdot 2^k - 2^k - 2^{k+1} = 0 \geq 0$$

   综合1，2，当 $i = 2^k$ 时有存款。

综上，任何操作后存款总额均不会出现负数。

总平摊代价为 $3n$ ，平摊代价为 $O(1)$

势能法

定义:

$\Phi(D_0) = 0$

$\Phi(D_i) = 2i - 2 \cdot 2^{\lfloor \log_2 i \rfloor} \geq 0$

- 若 $i \neq 2^k$,$c_i = 1 + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 2 = 3$
- 若 $i = 2^k$，$c_i = i + 1 + \Phi(D_i) - \Phi(D_{i-1}) = i + (2i - 2 \cdot 2^k) - (2(i-1) - 2 \cdot 2^{k-1}) = 3$

因此平摊代价为 $O(1)$