算法设计与分析第七章作业

毛翰翔　　　210110531

2022 年 11 月 1 日

1. Solution　聚集法:

对于 n 次操作, 其中开销不为 1 的操作次数为 $\lfloor \log_2 n \rfloor$, 这些操作的开销即为

$$\sum_{i=1}^{\lfloor \log n \rfloor} 2^i \le \sum_{i=0}^{\lfloor \log_2 n \rfloor} 2^i \le \frac{2^{\lfloor \log_2 n \rfloor} - 1}{2-1} \le n$$

那么总的开销为开销不为 1 的加上开销为 1 的部分, 其中开销为 1 的数量小于 n

$$\sum_{i=1}^{n} c_i \le n + n = 2n$$

即, $T(n) = O(n)$, 那么 $T(n)/n = O(1)$

会计法:　开销为 1 的操作的摊还代价为 3, 开销不为 1 的操作的摊还代价为 0. 每次进行非 1 的操作时, 设这次操作是第 i 次, 拿走前面 $i - i/2$ 次操作所给出的 credits. 而每一次 1 操作会给出 2 的 credits. 则非 1 操作的实际开销为

$$\frac{i}{2} \cdot 2 + 0 = i$$

和题意相符, 于是有 $T(n) \le 3n$, 即 $T(n) = O(n)$

势能法:　设当前操作数为 n , 设 $\Phi(D) = 2\left[n - 2^{\lfloor \log_2 n \rfloor}\right]$, 类似的, 1 操作的摊还代价为

$$\hat{c} = 1 + \Phi(D_{n+1}) - \Phi(D_n) = 3$$

非 1 操作的摊还开销为

$$\hat{c} = i + \Phi(D_{n+1}) - \Phi(D_n) = 0$$

那么 $T(n) = O(n)$, 并且 $T(n)/n = O(1)$

2. Solution　聚集法:　当进行了 n 次 `flip_push` 之后, 其中的反转操作次数为 $\lfloor \log_2 n \rfloor$, 其中每一次反转的操作的代价是 i, 设反转是 i 次操作类似的, 我们有

$$\sum_{k=1}^{\lfloor \log_2 n \rfloor} 2^k \le n$$

那么, 总的代价为

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

即 $T(n)/n = O(1)$

会计法:　设每次进栈的操作的摊还代价是 3, 而反转操作的摊还代价是 0 , 那么每一个栈元素上

的 credits 是 2. 反转的摊还代价为 0. 每次进栈时, 设这次是第 i 次, 拿走前面 $i-i/2$ 个栈元素上的 credits. 则反转的实际开销为

$$\frac{i}{2}\cdot 2+0=i$$

和题意相符, 于是有 $T\left(n\right)\leq 3n$, 即 $T\left(n\right)=O\left(n\right)$

聚能法: 设聚能函数 $\Phi : S\mapsto \Phi(S)=2\left|S\right|$, 其中 $\left|S\right|$ 代表 S 中没有被反转过的元素个数. 那么进栈操作的摊还代价是

$$\hat{c}=1+\Phi(S_{n+1})-\Phi\left(S_n\right)=3$$

反转操作的摊还代价为

$$\hat{c}=i+\Phi(S_{n+1})-\Phi\left(S_n\right)=i+2\left(0-\frac{i}{2}\right)=0$$

那么有 $T\left(n\right)=O\left(n\right)$