DSA_10

姓名：邵宁录　　　学号：2018202195

17.1-3

下面用 **聚合分析** 来计算该问题的摊还代价。

不妨设该操作序列共有 n 个操作，其中，有 $2^k \leq n < 2^{k+1}$

那么，该n次操作中，共有 $k+1$ 次操作的代价为 2^k ，其余 $n-k-1$ 次操作的代价为 1 。

于是乎，就有

$$
\begin{aligned}
总代价 &= \sum_{i=1}^{n-k-1} 1 + \sum_{i=0}^{k} 2^i \\
&= n-k-1+2^{k+1}-1 \\
&< n-k-1+2n-1 \\
&= 3n-k-2
\end{aligned}
$$

所以，对于一个由n个操作组成的操作序列，其代价为 $\mathrm{O}(n)$ 。因此，每个操作的平均代价，即摊还代价为 $\mathrm{O}(n)/n = \mathrm{O}(1)$

17.2-2

下面用 **核算法** 来计算该问题的摊还代价。

设当 i 为 2 的幂次方的时候，我们赋予其摊还代价为 2；其他时候，我们赋予其摊还代价为 3 。

我们可以知道，当 i 为 2 的幂次方的时候，其实际代价为 i；其他时候，其实际代价为 1 。

下面进行分类讨论：

1. 当 $i=1$ 时，我们支付 2 点代价，其中 1 点用于支付实际代价，1 点用于存储为信用。此时存储的信用为 1 。
2. 当 $i=2$ 时，我们支付 2 点代价，其中 2 点用于支付实际代价，0 点用于存储为信用。次时存储的信用为 1 。
3. 当 $i>2$时，若 i 不是 2 的幂次方，则其摊还代价一直为 3 ，实际代价一直为 1 ，此时每次都能有 2 点作为信用存储，满足核算法的条件。从 2^i 到 2^{i+1} 次操作，其中摊还代价为 1 的操作数量为 2^i-1 次，则可以累积 $2\times(2^1-1)=2^{i+1}-2$ ，再加上第 2^i+1 次其本身的摊还代价为 2 点，总共正好有 2^{i+1} 点累积的信用可以与该次的实际代价 2^{i+1} 抵消为零，因此也满足核算法的条件。

综上，核算法成立，该问题的每个操作的摊还代价为 $\mathrm{O}(1)$ 。

17.3-2

下面用 **势能法** 对问题求解。

定义：

$$
\Phi(D_i) = \begin{cases} 0, & i=0 \\ 2i-2^{1+\lfloor \lg i \rfloor}, & i>0 \end{cases}
$$

下面进行分类讨论：

1. 当 $i = 1$ 时，有 $\hat{c}_i = c_i + \Phi(D_1) - \Phi(D_0) = 1 + 2 - 2 - 0 = 0$ 。
2. 当 $i > 1$ 且 i 不是 2 的幂次方时，有 $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 2i - 2^{1+\lfloor \lg i \rfloor} - 2(i-1) + 2^{1+\lfloor \lg(i-1) \rfloor} = 3$ 。
3. 当$i > 1$ 且 i 是 2 的幂次方时，不妨设 $i = 2^k$ ，则有 $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = i + 2i - 2^{1+k} - 2(i-1) + 2^k = 3i + 2 - 2i - 2^k = 3i + 2 - 3i = 2$ 。

综上，即证。因此该问题的每个操作的摊还代价为 $\mathrm{O}(1)$ 。