# 数据结构与算法 I 作业 17

2019201409 于倬浩

2020 年 12 月 5 日

### 17.1-3 聚合分析

设 $n$ 次操作的总运行时间为 $T(n)$，则有：

$$\begin{aligned}
T(n) &= \sum_{i=1}^{n} [i = 2^k]i + [i \neq 2^k]1 \\
&= \sum_{i=1}^{\lfloor lg(n) \rfloor} 2^i + \sum_{i=1}^{n} 1 - \lfloor lg(n) \rfloor \\
&= \sum_{i=1}^{\lfloor lg(n) \rfloor} 2^i + n - lg(n) \\
&\leq 2n + n - lg(n) \\
&= 3n - lg(n) \\
&= O(n)
\end{aligned}$$

因此，单次操作的运行时间则为 $T(n)/n = O(n)/n = O(1)$。

### 17.2-2 核算分析

设第 $i$ 个操作的实际代价为 $c_i$，摊还代价为 $\hat{c_i}$，有：

$$c_i = \begin{cases} 1 & (i \neq 2^k) \\ i & (i = 2^k) \end{cases}$$

令 $\hat{c_i} = 3$，即令每个操作的摊还代价为 3。每个操作若实际代价为 1，则存起来 2 的 credit；如果实际代价大于 1，则开始消耗之前存起来的 credit。

根据 `17.1-3`，对任意 n 都有：

$$
\begin{aligned}
\sum_{i=1}^{n} c_i &\leq 3n - lg(n) \\
&\leq 3n \\
&\leq \sum_{i=1}^{n} \hat{c_i}
\end{aligned}
$$

因此 n 个操作的总代价为 $O(n)$。

## 17.3-2 势能分析

设 $D_i$ 表示第 $i$ 次操作后的状态，$\Phi(D_i)$ 为此状态的势函数。

首先，使用势能分析，要满足：

$$
\begin{aligned}
\hat{c_i} = c_i + \Phi(D_i) - \Phi(D_{i-1}) \\
\sum_{i=1}^{n} \hat{c_i} \geq \sum_{i=1}^{n} c_i
\end{aligned}
$$

因此，尝试构造 Φ 如下：

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

这种构造是利用势能分析的实际意义定义的。当 $i = 2^k$ 时，$\Phi(D_i) - \Phi(D_{i-1}) = i - 2^{k+1} - ((i-1) - 2^k) = 1 - 2^k = 1 - i$，而当 $i \neq 2^k$ 时，$\Phi(D_i) - \Phi(D_{i-1}) = 1$。这两点可以符合势能差的实际含义，但是并不能满足 $\Phi(D_i) \geq 0$ 恒成立，因此修改构造如下：

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

这样，可以保证势函数恒为正，势能差满足：

$$\Phi(D_i) - \Phi(D_{i-1}) = \begin{cases} 2 - i & i = 2^k \\ 2 & i \neq 2^k \end{cases}$$

因此有：

$$\begin{aligned} \sum_{i=1}^{n} \hat{c}_i &= \sum_{i=1}^{n} c_i + \Phi(D_i) - \Phi(D_{i-1}) \\ &= \sum_{i=1}^{n} [i = 2^k](2 - i + i) + [i \neq 2^k](2 + 1) \\ &\leq 3n \end{aligned}$$

因此，$n$ 次操作的运行时间为 $O(n)$，单次操作摊还代价为 $O(1)$。