Homework 8

PB17000297 罗晏宸

November 24 2019

1 Exercise 17.1-3

假定我们对一个数据结构执行一个由 n 个操作组成的操作序列，当 i 严格为 2 的幂时，第 i 个操作的代价为 i，否则代价为 1。使用聚合分析确定每个操作的摊还代价。

解 在执行的 n 个操作中，有至多 $\lceil \lg n \rceil$ 个操作的次序是 2 的幂，这些操作的次序（即代价）如下

$$1, 2, 4, 8, \cdots, 2^{\lceil \lg n \rceil}$$

n 个操作的总代价为

$$\begin{aligned}
T = &\sum_{k=0}^{\lceil \lg n \rceil} 2^k + (n - \lceil \lg n \rceil) \times 1 \\
= &2^{\lceil \lg n \rceil + 1} - 1 + (n - \lceil \lg n \rceil) \\
\leq &2^{\lg n + 2} + n - \lg n \\
= &3 \lg n + n
\end{aligned}$$

因此每个操作的摊还代价是 $O\left(\dfrac{3 \lg n + n}{n}\right) = O(1)$ 的。

2 Exercise 17.2-2

用核算法重做第一题。

解 当操作次序是 2 的幂时，为其赋 4 的摊还代价，否则为其赋 2 的摊还代价，则每一个不为 2 的幂的操作均会提供 1 的信用以支付差额，对于一个 n 个操作组成的操作序列，有

$$
\begin{aligned}
&4 \times \lceil \lg n \rceil + 2 \times (n - \lceil \lg n \rceil) \\
=&2 \times \lceil \lg n \rceil + 2n \\
\le&3 \lg n + n \\
\le&\sum_{k=0}^{\lceil \lg n \rceil} 2^k + (n - \lceil \lg n \rceil) \times 1
\end{aligned}
$$

每个操作的摊还代价都是常数，因此都是 $O(1)$ 的。

3 Exercise 17.3-2

使用势能法重做第一题。

解 定义势函数 $\Phi(D_i)$

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

则有对任意的 i，$\Phi(D_i) \ge 0$，且

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

故有

$$
\begin{aligned}
&\sum_{i=1}^{n} \hat{c}_i \\
=&\sum_{i=1}^{n} 1 \\
=&n
\end{aligned}
$$

因此每个操作的摊还代价是 $O(1)$。

4 Problem 30-3 Multidimensional fast Fourier transform

我们可以将一维离散傅里叶变换推广到 d 维上。这时输入是一个 d 维的数组 $A = (a_{j_1,j_2,\cdots,j_d})$，维数分别为 $n_1, n_2, \cdots, n_d$，其中 $n_1 n_2 \cdots n_d = n$。定义 d 维离散傅里叶变换如下：

$$y_{k_1,k_2,\cdots,k_d} = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} \cdots \sum_{j_d=0}^{n_d-1} a_{j_1,j_2,\cdots,j_d} \omega_{n_1}^{j_1 k_1} \omega_{n_2}^{j_2 k_2} \cdots \omega_{n_d}^{j_d k_d}$$

其中 $0 \le k_1 < n_1, 0 \le k_2 < n_2, \cdots, 0 \le k_d < n_d$。

a 证明：我们可以依次在每个维度上计算一维的 DFT 来计算一个 d 维的 DFT。也就是说，首先沿着第 1 维计算 n/n_1 个独立的一维 DFT。然后，把沿着第 1 维的 DFT 结果作为输入，我们计算沿着第 2 维的 n/n_2 个独立的一维 DFT。利用这个结果作为输入，我们计算沿着第三维的 n/n_3 个独立的一维 DFT，如此下去，直到第 d 维。

b 证明：维度的次序并无影响，于是可以通过在 d 个维度的任意顺序中计算一维 DFT 来计算一个 d 维的 DFT。

c 证明：如果采用计算快速傅里叶变换计算每个一维的 DFT，那么计算一个 d 维的 DFT 的总时间是 $O(n \lg n)$，与 d 无关。

解

a 可以交换求和号的顺序

$$\begin{aligned}
& y_{k_1,k_2,\cdots,k_d} \\
&= \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} \cdots \sum_{j_d=0}^{n_d-1} a_{j_1,j_2,\cdots,j_d} \omega_{n_1}^{j_1 k_1} \omega_{n_2}^{j_2 k_2} \cdots \omega_{n_d}^{j_d k_d} \\
&= \sum_{j_d=0}^{n_d-1} \cdots \sum_{j_2=0}^{n_2-1} \left(\sum_{j_1=0}^{n_1-1} a_{j_1,j_2,\cdots,j_d} \omega_{n_1}^{j_1 k_1} \right) \omega_{n_2}^{j_2 k_2} \cdots \omega_{n_d}^{j_d k_d}
\end{aligned}$$

故可以先计算 n/n_1 个独立的一维 DFT：$\sum\limits_{j_1=0}^{n_1-1} a_{j_1,j_2,\cdots,j_d}\omega_{n_1}^{j_1k_1}$，再将其结果（记为 $a^{(1)}_{j_1,j_2,\cdots,j_d}$）作为输入继续计算

$$\sum_{j_d=0}^{n_d-1}\cdots\sum_{j_3=0}^{n_3-1}\left(\sum_{j_2=0}^{n_2-1} a^{(1)}_{j_1,j_2,\cdots,j_d}\omega_{n_2}^{j_2k_2}\right)\omega_{n_3}^{j_3k_3}\cdots\omega_{n_d}^{j_dk_d}$$

直到第 d 维

$$\sum_{j_d=0}^{n_d-1} a^{(d-1)}_{j_1,j_2,\cdots,j_d}\omega_{n_d}^{j_dk_d}$$

b 如 a 中所述的计算过程实际上由于求和号与乘号的交换性，可以任意调换顺序。

c 对于 a 中的计算过程，沿着第 k 维的计算实际需要计算 $n\left/\prod\limits_{i\le k} n_k\right.$ 个独立的 DFT，因此消耗 $O\left(\lg n_k\cdot n\left/\prod\limits_{i\le k} n_k\right.\right)$ 的时间，总的时间

$$\begin{aligned}
T =& \sum_{k=1}^{d}\frac{n}{\prod\limits_{i\le k} n_k}\cdot\lg n_k \\
\le& \sum_{k=1}^{d}\frac{n}{\prod\limits_{i\le k} n_k}\cdot\lg n \\
\le& \sum_{k=1}^{d}\frac{n}{\prod\limits_{i\le k} 2}\cdot\lg n \\
\le& n\lg n\cdot\sum_{k=1}^{d}\frac{1}{2}
\end{aligned}$$

$$= O(n\lg n)$$

即，与 d 无关