算法基础第六次作业

艾语晨 PB18000227

2020 年 12 月 8 日

目录

1 **2**
1.1 聚合分析 2
1.2 核算法 2
1.3 势能法 2
1.4 多维快速傅立叶变换 3
(a) a. 3
(b) b. 3
(c) c. 3

第六次作业

第 1.1 题　聚合分析

> 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，请用聚合分析确定每个操作的摊还代价

设 c_i 是第 i 个操作第代价，则由题意，有:

$$c_i = \begin{cases} i & i\text{是 2 的幂} \\ 1 & otherwise \end{cases}$$

故 n 个操作的代价为:

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{\lceil \lg n \rceil} + \sum_{i \le n \text{ 且不是 2 的幂}} 1 \le \sum_{i=1}^{\lceil \lg n \rceil} +n = 2^{1+\lceil \lg n \rceil} - 1 + n \le 4n - 1 + n \le 5n \in O(n)$$

由于要求平均，对以上结果除以 n，得到时间复杂度为 $O(1)$

第 1.2 题　核算法

给每一个操作赋以代价 3。第一个操作有剩余的代价 2。现在假设我们在执行完第 2^i 步之后还有非负的剩余代价，而接下来的 $2^i - 1$ 步都会剩余代价 2，故加起来有 $2 \times (2^{i-1} - 1) = 2^{i+1} - 1$ 的剩余代价。而在执行完第 2^{i+1} 步之后，剩余代价为 $2^{i+1} - 1 + 2 - 2^{i+1} = 1$ 的剩余代价，依然非负。由数学归纳法可知，对任意步骤都有非负的代价。由于每一步的时间复杂度为 $O(1)$，故整体的摊还复杂度为 $O(n)$

第 1.3 题　势能法

当 $i = 2^k$ 时，令 $\Phi(D_i)k+3$，否则，令 k 为满足 $2^k \le i$ 的最大整数。则有 $\Phi(D_i) = \Phi(D_{2^k}) + 2(i-2^k)$，同时定义 $\Phi(D_0) = 0$。那么，对于所有的 $i \ge 0$，$\Phi(D_i) \ge 0$。势能差 $\Phi(D_i) - \Phi(D_{i-1})$ 在 $i = 2^k$ 时是 $-2^k + 3$，否则为 2，故 n 步操作的摊还代价为 $\sum_{i=1}^{n} \hat{c}_i = 3n = O(n)$

第 1.4 题 多维快速傅立叶变换

(a) a.

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

小括号里面的是一重傅立叶变换，而它需要对于每一个 a 的取值计算 $n_2 n_3 \cdots n_d = n/n_1$ 次。将内层计算之后向外归纳，可知对于第 k 维，要做 $n/\prod_{i<k} n_i$ 次 DFT

(b) b.

求和的顺序可以为任意，因为下标之间相互是独立的

(c) c.

在第 k 维做每一次 DFT 的代价是 $O(n_k \lg(n_k))$ 的，而它需要做总共 $n/\prod_{i<k} n_i$ 次，故总时间代价为 $O(n/\prod_{i<k} n_i \lg(n_k))$。

$$
\begin{aligned}
\sum_{k=1}^{d} n/\left(\prod_{i<k} n_i\right) \lg(n_k) &\le \lg(n) \sum_{k=1}^{d} n/\left(\prod_{i<k} n_i\right) \\
&\le \lg(n) \sum_{k=1}^{d} n/2^{k-1} \\
&< n \lg n
\end{aligned}
$$

与 d 独立