Homework 8

PB17111623 范睿

2019 年 12 月 8 日

0.1 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。

$$totalcost = \sum_{i=1}^{n} cost(i) \le \sum_{i=1}^{n} 1 + \sum_{i=0}^{\lfloor \log_2 n \rfloor} (2^i - 1) = n + (2n-1) - (\lfloor \log_2 n \rfloor + 1) = 3n - \lfloor \log_2 n \rfloor - 2$$
$$averagecost = \mathcal{O}(\frac{3n - \lfloor \log_2 n \rfloor - 2}{n}) = \mathcal{O}(1)$$

0.2 用核算法重做第一题。

重新定义每次操作的代价：
当 i 严格为 2 的幂时，第 i 个操作的代价为 2*i，其余代价为0。
新定义的代价下，n次操作总代价一定比题目中总代价大。在新定义的代价下：

$$newtotalcost = \sum_{i=0}^{\log_2 n} (2 \times 2^i) = 4n - 2 \ge 3n - \lfloor \log_2 n \rfloor - 2 = totalcost$$

在新定义的代价下，平均代价为：

$$averagecost = \mathcal{O}(\frac{4n-2}{n}) = \mathcal{O}(1)$$

0.3 使用势能法重做第一题。

定义$D(0) = 0,\ D(i) = 2i - 2^{\lfloor \log_2 i \rfloor}$
$\forall n > 0, D(n) > 0$
在D(i)如此定义下，$c1' = c1 + D(1) - D(0) = 1$
$\forall i$不是2的整数次幂，$ci' = ci + D(i) - D(i-1) = 3$
$\forall i$是2的整数次幂，
$ci' = ci + D(i) - D(i-1) = i + 2i - 2^{1+\lfloor \log_2 i \rfloor} - (2(i-1) - 2^{1+\lfloor \log_2 i \rfloor - 1}) = i + 2 - 2^{1+\lfloor \log_2 i \rfloor - 1} = 3$
因此

$$newtotalcost = 1 + \sum_{i=2}^{n} 3 = 3n - 2$$
$$averagecost = \mathcal{O}(1)$$

0.4

0.4.1 a

首先证明，利用沿着第1维计算n/n1个独立的一维DFT的结果可以计算出前两维的DFT：

沿着第一维计算的DFT的结果为

$$y_{k_1,j_2,...,j_d} = \sum_{j_1=0}^{n_1-1} a_{j_1,j_2,...,j_d}\omega_{n_1}^{j_1k_1}$$

对$y_{k_1,j_2,...,j_d}$在第二维上进行DFT，结果为

$$y_{k_1,k_2,...,j_d} = \sum_{j_2=0}^{n_2-1} y_{k_1,j_2,...,j_d}\omega_{n_2}^{j_2k_2} = \sum_{j_2=0}^{n_2-1}(\sum_{j_1=0}^{n_1-1} a_{j_1,j_2,...,j_d}\omega_{n_1}^{j_1k_1})\omega_{n_2}^{j_2k_2} = \sum_{j_2=0}^{n_2-1}\sum_{j_1=0}^{n_1-1} a_{j_1,j_2,...,j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}$$

因此，有第1维的DFT结果可得前两维的DFT结果。
下面证明，由前i为的DFT结果可得前i+1维的DFT结果：
前i维的DFT结果为：

$$y_{k_1,k_2,...,k_i,j_{i+1},...,j_d} = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1}...\sum_{j_i=0}^{n_i-1} a_{j_1,j_2,...j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}...\omega_{n_i}^{j_ik_i}$$

对$y_{k_1,k_2,...,k_i,j_{i+1},...,j_d}$进行第i+1维的DFT，得到：

$$y_{k_1,k_2,...,k_i,k_{i+1},j_{i+2}...,j_d} = \sum_{j_{i+1}=0}^{n_{i+1}-1} y_{k_1,k_2,...,k_i,j_{i+1},...,j_d}\omega_{n_{i+1}}^{j_{i+1}k_{i+1}} = \sum_{j_{i+1}=0}^{n_{i+1}-1}(\sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1}...\sum_{j_i=0}^{n_i-1} a_{j_1,j_2,...j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}...\omega_{n_i}^{j_ik_i})\omega_{n_{i+1}}^{j_{i+1}k_{i+1}} = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1}...\sum_{j_i=0}^{n_i-1}\sum_{j_{i+1}=0}^{n_{i+1}-1} a_{j_1,j_2,...j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}...\omega_{n_i}^{j_ik_i}\omega_{n_{i+1}}^{j_{i+1}k_{i+1}}$$

由数学归纳法可知，d维的傅里叶变换可以由再各个维度上分别做傅里叶变换得到。

0.5 b

我们可以用$y_{k_1,k_2,...,k_i,k_{i+1},...,k_d} = \sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1}...\sum_{j_d=0}^{n_d-1} a_{j_1,j_2,...j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}...\omega_{n_d}^{j_dk_d}$中求和的顺序来表示以维度为单位计算d维DFT的顺序，比如，在此公式中，我们认为计算d维DFT的顺序是：第1维，第2维，...，第d维。那么对于∀1-d的一个排序$(s_1, s_2, ...s_d)$，显然

$$\sum_{j_1=0}^{n_1-1}\sum_{j_2=0}^{n_2-1}...\sum_{j_d=0}^{n_d-1} a_{j_1,j_2,...j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}...\omega_{n_d}^{j_dk_d} = \sum_{j_{s_1}=0}^{n_{s_1}-1}\sum_{j_{s_2}=0}^{n_{s_2}-1}...\sum_{j_{s_d}=0}^{n_{s_d}-1} a_{j_1,j_2,...j_d}\omega_{n_{s_1}}^{j_{s_1}k_{s_1}}\omega_{n_{s_2}}^{j_{s_2}k_{s_2}}...\omega_{n_{s_d}}^{j_{s_d}k_{s_d}}$$

因此，可以证明，顺序对求和无影响。

0.6 c

由a知，计算第i维独立的DFT所用的时间为$\frac{n}{n_i}\mathcal{O}(n_i \lg n_i) = \mathcal{O}(n \lg n_i)$
因此，计算所有维的DFT所用的时间为$\mathcal{O}(\sum\limits_{i=1}^{d}(n \lg n_i)) = \mathcal{O}(n \lg n)$