HW6

PB19111713钟颖康

1.

(1)聚合分析

考虑n次操作的代价：

$$\begin{aligned} n-\lfloor log_2 n\rfloor-1+\sum_{k=0}^{\lfloor log_2 n\rfloor}2^k &= n-\lfloor log_2 n\rfloor-1+2^{\lfloor log_2 n\rfloor+1}-1 \\ &\le n-\lfloor log_2 n\rfloor-2+2n \\ &= 3n-2-\lfloor log_2 n\rfloor \\ &= O(n) \end{aligned}$$

故平均代价为$O(1)$

(2)核算法

给每次操作赋予3费用，

①第2^0次操作的代价是1，操作结束后信用为2

②假设第2^i次操作结束后信用非负，$i\ge 0$

③考虑第2^i+1到第2^{i+1}这一段操作，

　第2^i+1到第$2^{i+1}-1$这一段每次操作代价均为1，每次操作可以获得2点信用，总计$2^{i+1}-2$点信用；

　第2^{i+1}次操作费用为3，代价为第2^{i+1}；

　第2^i+1到第2^{i+1}这一段操作总计获得1点信用，即第2^{i+1}次操作结束后信用非负

综合①②③，对于每次操作都有非负信用。

故总有摊还代价为实际代价的上界，总代价为$O(3n)=O(n)$，平均代价为$O(1)$

(3)势能法

设$\phi(D_i)=\lfloor log_2 n\rfloor+3$，

令$\phi(D_0)=0,\phi(D_i)=\phi(D_{2^k})+2(i-2^k)$，则对任意$i$均有$\phi(D_i)\ge 0$，

若i不是2的幂次，则$\phi(D_i)-\phi(D_i-1)$的势差为2；

若i是2的幂次，则$\phi(D_i)-\phi(D_i-1)$的势差为$3-i$，

从而$\sum_{i=1}^{n}\widehat{c_i}=3n=O(n)$，平均代价为$O(1)$

2.

根据题意，$68^{k_1}=132464,70^{k_2}=143640,72^{k_3}=155424$,

计算可知$k_2\le k_1\le k_3$且$k_2\le log_2 7$，故第二种方法会得到最佳渐进时间且比$Strassen$算法更优

3.

a.

$$\begin{aligned}
y_{k_1,\ldots,k_d} &= \sum_{j_1=0}^{n_1-1}\cdots\sum_{j_d=0}^{n_d-1} a_{j_1,\ldots,j_d}\omega_{n_1}^{j_1k_1}\cdots\omega_{n_d}^{j_dk_d} \\
&= \sum_{j_d=0}^{n_d-1}\cdots\sum_{j_2=0}^{n_2-1}\sum_{j_1=0}^{n_1-1} a_{j_1,\ldots,j_d}\omega_{n_1}^{j_1k_1}\omega_{n_2}^{j_2k_2}\cdots\omega_{n_d}^{j_dk_d} \\
&= (\sum_{j_d=0}^{n_d-1}\cdots(\sum_{j_2=0}^{n_2-1}(\sum_{j_1=0}^{n_1-1} a_{j_1,\ldots,j_d}\omega_{n_1}^{j_1k_1})\omega_{n_2}^{j_2k_2})\cdots\omega_{n_d}^{j_dk_d})
\end{aligned}$$

对于第一维只需要计算$\frac{n}{n_1}$个独立一维DFT;

对于第二维，第一维计算出的每个结果都需要计算$\frac{n}{n_1n_2}$个独立的一维DFT，总共$\frac{n}{n_2}$个;

依此类推直到第d维，故可以。

b.

由于d次求和的上下界互相独立，故求和顺序可以任意调换，其余同**a.**，故可以。

c.

每个维度都是$O(n_i lg\, n_i)$，计算$\frac{n}{\prod_{j\le i} n_j}$次

$$\begin{aligned}
\sum_{i=1}^{d}\frac{n}{\prod_{j<i} n_j} lg\, n_i &\le lg(n)\sum_{i=1}^{d}\frac{n}{\prod_{j<i} n_j} \\
&\le lg(n)\sum_{i=1}^{d}\frac{n}{2^i-1} \\
&< n\, lg(n)
\end{aligned}$$

故总时间为$O(nlgn)$，与d无关。