HW8

王嵘晟　PB1711614

1.

对于这个由 n 个操作组成的操作序列，当且仅当 i 为 2^m，$m \in N$ 时第 i 个操作的代价为 i，其他代价为 1. 所以这个操作序列的总个数为 n 时，设 $2^k \leq n < 2^{k+1}$，则执行 n 个操作的序列的时间为 $(n-k) \times 1 + \frac{2 \times (1-2^k)}{1-2} = 2^{k+1} - 2 + n - k$，因为 $k = \lfloor log(n) \rfloor$，所以总时间为 $O(n)$，即摊还代价为 $O(n)/n = O(1)$

2.

令每个操作的摊还代价为 3，令第一个操作的实际代价为 1，则信用为 2。假设执行第 2^i 个操作后的信用为非负的，接下来的 $2^i - 1$ 个操作的实际代价为 1，则可从每个操作获得的信用都为 2。总信用为 $2^{i+1} - 2$。对于第 2^{i+1} 个操作，一开始设定的 3 个代价给了 $2^{i+1} + 1$ 个信用来使用，而代价为 2^{i+1}，仍然留下了一个信用。所以对于任意一个操作，信用都是非负的。所以对于每个操作来说，摊还代价为 $O(1)$，n 个操作的总摊还代价为 $O(n)$。

3.

当操作序号 $i = 2^k$ 时，令 $\Phi(D_i) = k+3$，否则 $k = \lfloor log(i) \rfloor$，令 $\Phi(D_i) = \Phi(D_{2^k}) + 2(i - 2^k)$。令 $\Phi(D_0) = 0$，则对于 $\forall i \geq 0$，$\Phi(D_i) \geq 0$。当 i 或 i-1 都不等于 2^k 时，$\Phi(D_i) - \Phi(D_{i-1}) = 2$。当 $i = 2^k$ 时，$\Phi(D_i) - \Phi(D_{i-1}) = 0$，所以 n 个操作的总摊还代价为 $\sum_{i=1}^{n} \hat{c}_i = O(n)$。

4.

4.(a)

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

括号中的内容即为一维的 DFT,由于需要给求和的每一项都计算,所以一共要计算 $n_2n_3...n_d = n/n_1$ 次，由于每一项 a 的值可能不同，所以每计算一次，求和的总数目减一。通过这个方法可以继续减少维数，对第 k 维来说，需要计算 $n/(\prod_{i\le k} n_i)$ 个独立的一维 DFT。

4.(b)

对于求和，由于任何被求和的元素都没有出现在求和边界上。所以求和的顺序可以随意交换，维度的次序不会影响。

4.(c)

对运算每个 DFT 来说，第 k 维的时间为 $O(n_klog(n_k))$。只需要执行 $n/(\prod_{i\le k} n_i)$ 次，所以总时间为 $O(n/(\prod_{i\le k} n_i)log(n_k))$
$\sum_{k=1}^{d} n/(\prod_{i\le k} n_i)log(n_k) \le lgn\sum_{k=1}^{d} n/(\prod_{i\le k} n_i) \le lgn\sum_{k=1}^{d} n/(2^{k-1}) < nlgn$
所以总时间为 $O(nlgn)$