

小波变换

by Tianpeng Li

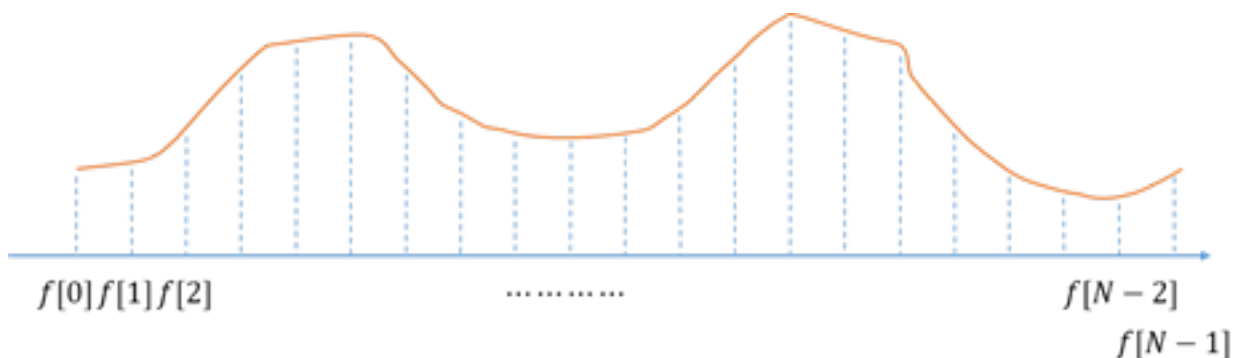
时间限制: 1000 ms

内存限制: 3000 KB

问题描述

小波变换是信号处理中一种常用的信号处理方法。虽然小波变换的理论比较抽象，但是小波变换方法却很容易操作，并且小波变换的程序是可以用递归的方法方便快速的实现。通过这个实验，可以帮助大家了解递归程序的编写与递归方法在实际问题中的应用。

在这个实验中，我们将应用Haar小波 (https://en.wikipedia.org/wiki/Haar_wavelet) 来对一个一维的函数进行小波变换。在计算机中，一个一维的函数 $f(x)$ 可以用一组离散的数值来描述，比如 $f[0], f[1], \dots, f[N-1]$ ，如下图所示，这个函数在计算机中可以用一个一维数组来描述。使问题简单化，我们假设 $N=2^L, L=1, 2, 3, \dots$



最简单的Haar小波变换由下面 L 次计算获得：>

1) 第1次计算，

$$f_1[n] = \begin{cases} \frac{(f[2n] + f[2n + 1])}{2}, & n = 0, 1, \dots, \frac{N}{2} - 1 \\ -\frac{(f[2(n - \frac{N}{2})] - f[2(n - \frac{N}{2}) + 1])}{2}, & n = \frac{N}{2}, \frac{N}{2} + 1, \dots, N - 1 \end{cases}$$

2) 第2次计算，

$$f_2[n] = \begin{cases} \frac{(f_1[2n] + f_1[2n+1])}{2}, n = 0, 1, \dots, \frac{N}{4} - 1 \\ -\frac{(f_1[2(n - \frac{N}{4})] - f_1[2(n - \frac{N}{4}) + 1])}{2}, n = \frac{N}{4}, \frac{N}{4} + 1, \dots, \frac{N}{2} - 1 \\ f_1[n], n = \frac{N}{2}, \frac{N}{2} + 1, \dots, N - 1 \end{cases}$$

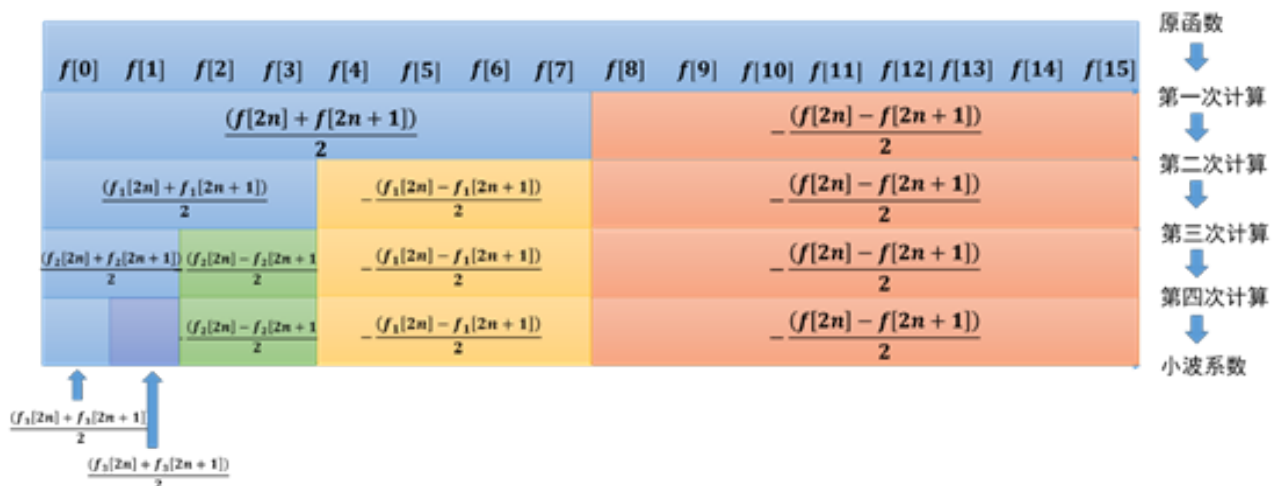
3) 第3次计算,

$$f_3[n] = \begin{cases} \frac{(f_2[2n] + f_2[2n+1])}{2}, n = 0, 1, \dots, \frac{N}{8} - 1 \\ -\frac{(f_2[2(n - \frac{N}{8})] - f_2[2(n - \frac{N}{8}) + 1])}{2}, n = \frac{N}{8}, \frac{N}{8} + 1, \dots, \frac{N}{4} - 1 \\ f_2[n], n = \frac{N}{4}, \frac{N}{4} + 1, \dots, N - 1 \end{cases}$$

4) 依次类推, 可以得出, 第m次计算

$$f_m[n] = \begin{cases} \frac{(f_{m-1}[2n] + f_{m-1}[2n+1])}{2}, n = 0, 1, \dots, \frac{N}{2^m} - 1 \\ -\frac{(f_{m-1}[2(n - \frac{N}{2^m})] - f_{m-1}[2(n - \frac{N}{2^m}) + 1])}{2}, n = \frac{N}{2^m}, \frac{N}{2^m} + 1, \dots, \frac{N}{2^{m-1}} - 1 \\ f_{m-1}[n], n = \frac{N}{2^{m-1}}, \frac{N}{2^{m-1}} + 1, \dots, N - 1 \end{cases}$$

在上面的计算中, $m=1, 2, \dots, L$, 即共进行L次运算, 既可以得到一个函数的小波变换。下图简单解释了对长度为16的数组进行小波变换的过程



请根据上面的算法, 编写小波变换的运算程序, 实现对长度为 $2L$ 的一维数组的小波变换。

输入格式

第一行一个正整数即 $L \leq 25$ ，信号长度为 2^L

第二行 2^L 个浮点数，用空格隔开，即是待变换信号

输出格式

一行 2^L 个浮点数，用空格隔开表示变换后信号，只要误差在0.001以内都算正确，可以选择保留4位小数。

输入样例

```
4
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

输出样例

```
8.5000 4.0000 2.0000 2.0000 1.0000 1.0000 1.0000 1.0000  
0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000 0.5000
```

提示

本次作业后面的样例 L 可能较大，采用递归算法时需要考虑到内存的使用多少，否则可能会超出内存限制。