FWT

简介

FWT快速沃尔什变换），是用来处理将按位运算作为操作符的多项式卷积，以及其衍生出来的一些问题。

在接触之前，你**必须**已经基本掌握FFT快速傅里叶变换），两者的原理内核基本一致。

算法内容

大概思路

FWT的基础功能，是用来处理下面的求和式子 $$ \begin{array}{l} C_{k}=\sum_{i | j=k} A_{i} * B_{j} \\ C_{k}=\sum_{i \& j=k} A_{i} * B_{j} \\ C_{k}=\sum_{i \wedge j=k} A_{i} * B_{j} \end{array} $$ 这样的式子虽然看似卷积，但没有办法用FFT来做。

我们先想想FFT的工作：先对于一个多项式求出他在若干个单位根的点值表示法，再将多项式乘起来，最后再复原。那么我们实际上也可以有类似的思路：能否先将多项式求出另外一个多项式$\mathrm{FWT}(A)$再将对应的位置乘起来，最后再复原？

也就是 $$ \mathrm{F W T}(C)=\mathrm{F W T}(A) * \mathrm{F W T}(B) $$ 继续讨论前，我们先定义以下的东西 $$ \begin{aligned} &A+B=\left(A_{0}+B_{0}, A_{1}+B_{1}, \dots \dots\right)\\ &\begin{array}{l} A-B=\left(A_{0}-B_{0}, A_{1}-B_{1}, \ldots \ldots\right) \\ A \oplus B=\left(\sum_{i \oplus j=0} A_{i} * B_{j}, \sum_{i \oplus j=1} A_{i} * B_{j}, \ldots \ldots\right) \end{array} \end{aligned} $$

或（or）卷积

或卷积写成向量的形式 $$ A | B=\left(\sum_{i | j=0} A_{i} * B_{j}, \sum_{i | j=1} A_{i} * B_{j}, \ldots \ldots\right) $$ 很显然，这个式子满足交换律 $$ A|B=B| A $$ 事实上也满足分配律 $$ (A+B) | C=\left(\sum_{i | j=0}\left(A_{i}+B_{i}\right) * C_{j}, \ldots \ldots\right) $$ 很显然可以把括号拆开分成两个$\sum$变成$A|C+B|C$

定义1：我们这样定义FWT映射 $$ \mathrm{F W T}(A)_i=\sum_{j | i=i} A_j $$ 这个式子可以靠感性理解其成立，严谨证明如下，下面$j\in i$代表$j|i=i$ $$ \begin{aligned} \mathrm{F W T}(C)_i &=\sum_{j \in i} C_j \\ &=\sum_{j \in i} \sum_{x | y=j} A_x \times B_y \\ &=\sum_{(x | y) \in i} A_x \times B_y \\ &=\sum_{x \in i} A_i \times \sum_{y \in i} B_i \\ &=\mathrm{F W T}(A)_i \times \mathrm{F W T}(B)_i \end{aligned} $$ **定义2：**等价定义为，对于一个多项式A最高次项为2^n我们把它分成两部分A_0,A_1分别表示前面2^{n-1}次项和后面2^{n-1}次项，也就是下标最高位为0与1两个部分 $$ \mathrm{F W T}(A)=\left\{\begin{array}{ll} \left(\mathrm{F W T}\left(A_{0}\right), \mathrm{F W T}\left(A_{0}+A_{1}\right)\right) & n>0 \\ A & n=0 \end{array}\right. $$ 其中的括号与逗号表示多项式拼接。

两者的等价性证明从简，其实是容易看出，如果定义2前式右边是$\mathrm{FWT}(A_0)+\mathrm{FWT}(A_1)$那么定义1是满足定义2的，而定义2

的FWT唯一，所以两者等价。也就是我们还需要证明一个等式（映射保加法，同态） $$ \mathrm{F W T}(A+B)=\mathrm{F W T}(A)+\mathrm{F W T}(B) $$ 证明如下 $$ \begin{array}{l} \mathrm{F W T}(A)_i=\sum_{j | i} A_j \\ \mathrm{F W T}(B)_i=\sum_{j | i} B_j \\ \mathrm{F W T}(A+B)_i=\sum_{j | i}(A+B)_j=\sum_{j | i} (A_j+B_j) \end{array} $$ 因此我们得到两个定义都是满足条件的。这里附带一个多余的步骤增强理解：通过定义2来导出$\mathrm{F W T}(C)=\mathrm{F W T}(A) * \mathrm{F W T}(B)$ $$ \begin{aligned} \mathrm{FWT}(A | B) &=\mathrm{FWT}\left((A | B)_{0},(A | B)_{1}\right) \\ &=\mathrm{FWT}\left(A_{0}\left|B_{0}, A_{0}\right| B_{1}+A_{1}\left|B_{0}+A_{1}\right| B_{1}\right) \\ &=\left(\mathrm{FWT}\left(A_{0} | B_{0}\right), \mathrm{FWT}\left(A_{0}\left|B_{0}+A_{0}\right| B_{1}+A_{1}\left|B_{0}+A_{1}\right| B_{1}\right)\right) \\ &=\left(\mathrm{FWT}\left(A_{0}\right) \times \mathrm{FWT}\left(B_{0}\right), \mathrm{FWT}\left(A_{0}\right) \times \mathrm{FWT}\left(B_{0}\right)+\mathrm{FWT}\left(A_{0}\right) \times \mathrm{FWT}\left(B_{1}\right)\\+\mathrm{FWT}\left(A_{1}\right) \times \mathrm{FWT}\left(B_{0}\right)+\mathrm{FWT}\left(A_{1}\right) \times \mathrm{FWT}\left(B_{1}\right)\right) \\ &=\left(\mathrm{FWT}\left(A_{0}\right) \times \mathrm{FWT}\left(B_{0}\right),\left(\mathrm{FWT}\left(A_{0}\right)+\mathrm{FWT}\left(A_{1}\right)\right) \times\left(\mathrm{FWT}\left(B_{0}\right)+\mathrm{FWT}\left(B_{1}\right)\right)\right) \\ &=\left(\mathrm{FWT}\left(A_{0}\right), \mathrm{FWT}\left(A_{0}+A_{1}\right)\right) \times\left(\mathrm{FWT}\left(B_{0}\right), \mathrm{FWT}\left(B_{0}+B_{1}\right)\right) \\ &=\mathrm{FWT}(A) \times \mathrm{FWT}(B) \end{aligned} $$

这实际上是一个数学归纳法的证明过程。证明$\mathrm{FWT}(A|B)=\mathrm{FWT}(A)*\mathrm{FWT}(B)$时，应用了$\mathrm{FWT}(A_i|B_j)=\mathrm{FWT}(A_i)*\mathrm{FWT}(B_j)$

与（and）卷积

与运算卷积的导出逻辑几乎和或运算一致，这里将大篇幅证明略去，仅仅给出结论 $$ \mathrm{FWT}(A)_i=\sum_{j\&i=i}A_j $$

$$ \mathrm{F W T}(A)=\left\{\begin{array}{ll} \left(\mathrm{F W T}\left(A_{0}+A_{1}\right), \mathrm{F W T}\left(A_{1}\right)\right) & n>0 \\ A & n=0 \end{array}\right. $$