# Задание №7
# Быстрое преобразование Фурье

### Описание алгоритма

Договоримся, что дискретное преобразование Фурье (ДПФ) мы будем записывать в виде:

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{\mp \frac{2\pi i}{N} nk},$$

где знак «−» соответствует прямому преобразованию, «+» — обратному.

Очевидно, что при прямой реализации ДПФ количество необходимых действий будет иметь порядок $O(N^2)$. Однако существуют алгоритмы, позволяющие существенно улучшить этот результат. Получим один из таких алгоритмов (т.н. случай «прореживания по частоте»).

Пусть $N$ представимо в виде $N = 2^n$. Запишем прямое дискретное преобразование Фурье (опустив для простоты коэффициент $1/\sqrt{N}$) в виде:

$$y_k = \sum_{n=0}^{N-1} x_n W_N^{nk}, \quad k = 0, \dots, N-1$$

где

$$W_p^q = e^{\frac{\mp 2\pi i q}{p}}.$$

Тогда

$$y_k = \sum_{n=0}^{N/2-1} \left( x_n W_N^{nk} + x_{n+N/2} W_N^{(n+N/2)k} \right) = \sum_{n=0}^{N/2-1} \left( x_n W_N^{nk} + W_N^{kN/2} x_{n+N/2} W_N^{nk} \right).$$

Поскольку

$$W_N^{kN/2} = (-1)^k,$$

то

$$y_k = \sum_{n=0}^{N/2-1} \left( x_n + (-1)^k x_{n+N/2} \right) W_N^{nk}.$$

Рассмотрим отдельно случаи для четных и нечетных значений $k$, представив $k$ как $2j$ или $2j+1$ соответственно ($j$ меняется от 0 до $N/2-1$). Тогда

$$y_{2j} = \sum_{n=0}^{N/2-1} \left( x_n + x_{n+N/2} \right) W_N^{2nj},$$

$$y_{2j+1} = \sum_{n=0}^{N/2-1} \left( x_n - x_{n+N/2} \right) W_N^{n(2j+1)}.$$

Заметим, что

$$W_N^{2nj} = W_{N/2}^{nj}, \quad W_N^{n(2j+1)} = W_{N/2}^{nj} \cdot W_N^n,$$

и получаем окончательно (заменяя индекс $j$ на $k$ для наглядности):

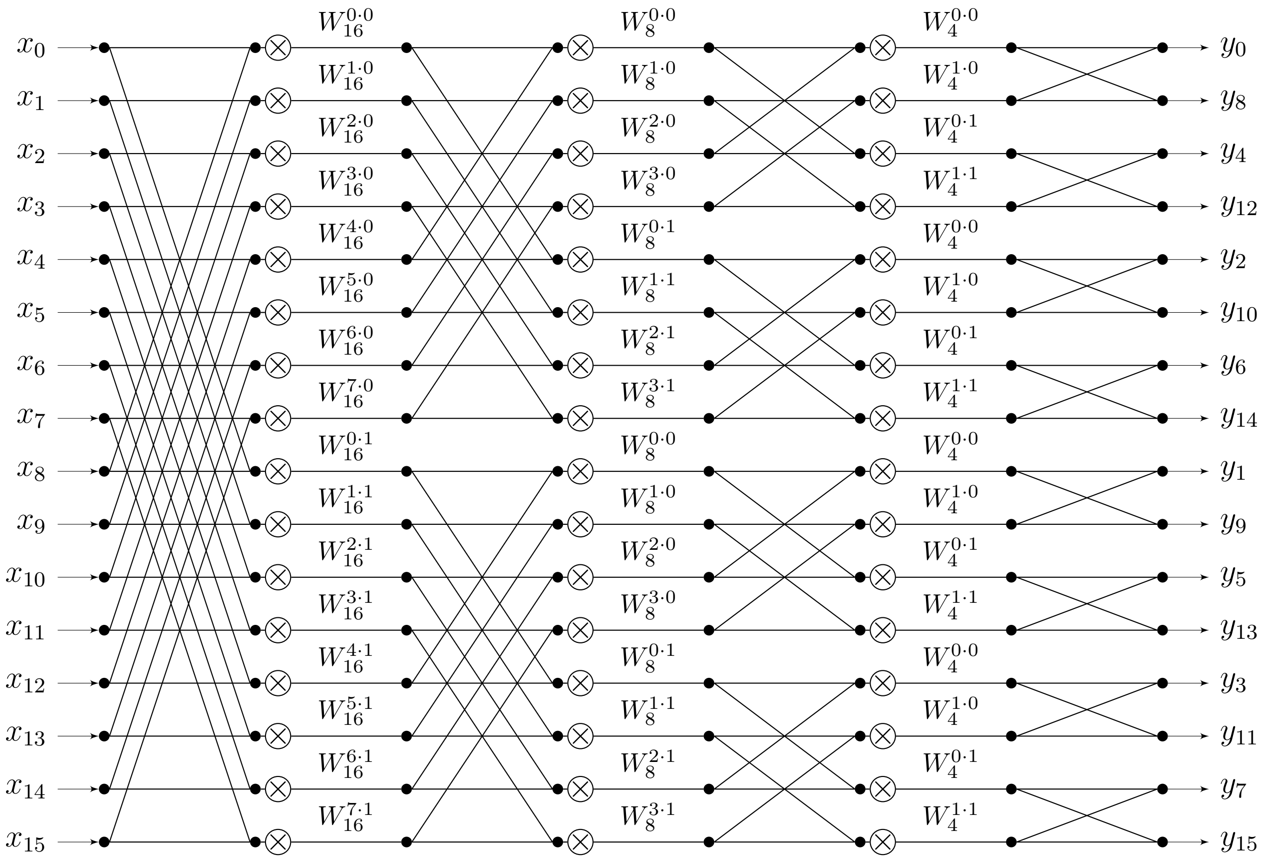
$$y_{2k} = \sum_{n=0}^{N/2-1} \left( x_n + x_{n+N/2} \right) W_{N/2}^{nk},$$

$$y_{2k+1} = \sum_{n=0}^{N/2-1} \left[ W_N^n \left( x_n - x_{n+N/2} \right) \right] W_{N/2}^{nk},$$

где $k = 0, \ldots, N/2-1$.

Видно, что оба выражения считаются фактически так же, как исходное, однако число слагаемых в каждой сумме уменьшилось в два раза. Повторяя аналогичную процедуру подсчета к каждой из этих сумм, можно добиться того, чтобы число слагаемых в каждом случае было невелико (обычно в качестве минимального значения $N$ берут 2 или 4), и вычислить такие суммы непосредственно.

Схема алгоритма для $N = 16$ представлена ниже:




В этом варианте алгоритма на каждом шаге имеющиеся данные делятся на два отрезка, а четные и нечетные спектральные значения вычисляются отдельно. Существует также второй вариант алгоритма (т.н. «прореживание по времени»), в котором, наоборот, из данных выбираются величины с четными и нечетными индексами, а спектральные индексы получаются в виде двух упорядоченных отрезков. С формальной точки зрения оба алгоритма не только эквивалентны, но и имеют одинаковую сложность, однако на практике рассмотренный выше вариант, как правило, несколько предпочтительнее (в силу особенностей архитектуры современных компьютеров, позволяющей получить существенный выигрыш в быстродействии при последовательном чтении данных из большой области памяти).

### Представление данных и результатов

Требуется реализовать алгоритм БПФ в виде функции или процедуры, выполняющей и прямое, и обратное преобразования (по некоторому параметру-ключу), а также тестирующую программу, использующую эту функцию. Исходные данные находятся в текстовом файле *data.dat*, в первой строке которого после символа ♯ и одного или нескольких пробелов находится $N$, а в последующих строчках пары чисел, разделенные одним или несколькими пробелами — действительные и мнимые части данных (если $N$ не является степенью двойки, то данные следует дополнить нулевыми значениями). Вывод результатов — в файл *result.dat* в таком же формате. Кроме этого, в файл *abs.dat* требуется вывести модули результата (по одному значению в строке, без предварительного вывода количества точек). Имейте в виду, что $N \sim 10^6 \div 10^7$, поэтому хранение промежуточных данных следует организовать по возможности экономно. Как обычно, желательно постараться сделать насколько возможно быстрый код.