Сортировка представляет собой процесс на каждом шаге которого, массив перезаписывается в другой массив, немного при этом упорядочиваясь. На первом шаге массив разбивается на пары и упорядочение выполняется внутри пар. На каждом последующем шаге в новые пары объединяются группы уже отсортированные на предыдущем шаге. Не трудно заметить, что группы каждого шага сортировки содержат количество элементов равное степени двойки. На нулевом шаге объединяемые группы содержат по 1 элементу, на первом по 2, на втором по 4 и т.д. Очень важно понять технику слияния групп. Пусть например, слиянию подлежат две группы по 4 элемента. Заметим, что эти группы могут быть получены двумя шагами сортировки и поэтому они уже упорядочены.

Пусть четверки таковы: (2, 7, 11, 25) (3, 5, 8, 36). Наша задача переписать их в массив из восьми элементов за один проход. Введем для каждой четверки указатели и установим их на первые элементы четверок. Элемент на который показывает указатель назовем Очередным. Тогда Очередных элементов у нас два Очередной1 и Очередной2. Заполнение очередной восьмерки массива - результата будет выполнятся следующей операцией:

Если Очередной1 > Очередного2
 То
  В массив - результат записывается Очередной1
  Указатель1 смещается вправо
 Иначе
  В массив - результат записывается Очередной2
  Указатель2 смещается вправо

Ниже показан процесс обработки нашего примера:
Элементы на которые показывают указатели будем помечать подчеркиванием

Первая четверка	***Вторая четверка***	***Результат***
2 7 11 25	3 5 8 36	2
2 **7** 11 25	3 5 8 36	2 3
2 7 11 25	3 5 8 36	2 3 5
2 7 11 25	3 5 8 36	2 3 5 7
2 7 11 25	3 5 8 36	2 3 5 7 8
2 7 11 25	3 5 8 36	2 3 5 7 8 11
2 7 11 25	3 5 8 36	2 3 5 7 8 11 25
2 7 11 25	3 5 8 36	2 3 5 7 8 11 25 36

Наверное не вызывает сомнение, что данный алгоритм работает быстрее чем например пузырьковая, а минус метода в требовании довольно большого объема дополнительной памяти.

Алгоритм

Данные:
Исходный массив
Массив результат

РАЗМЕР ИНТЕРВАЛА = 2

Пока РАЗМЕР ИНТЕРВАЛА < = Длина массива делать
  Начало
    Для каждой пары соседних интервалов (кроме последней пары) делать (пары: 1 и 2; 3 и 4 и т.д.)
        Слияние(Первый интервал пары, Второй интервал пары)
    Слияние(Предпоследний интервал, Остаток массива)
    Исходный массив = Массив результат // массив переписывается в массив
    РАЗМЕР ИНТЕРВАЛА=РАЗМЕР ИНТЕРВАЛА * 2
  Конец

Примечание. Операция перезаписи массива в массив может отнять большое время. Этого можно избежать если передавать не массив а адрес, а переменные работающие с массивами объявить как указатели. Этой же проблемы можно избежать если описать рекурсивную процедуру которая будет заниматься слиянием очередной массива в новый, а этот новый будет собственным массивом процедуры, но такая конструкция повлечет за собой существенные издержки памяти.

Процедура Слияние

Процедура получает на вход два интервала, из которых должна составить Новый интервал. Для этого сравниваются очередные числа входящих интервалов и на основании результата сравнения принимается решение, какое из них станет очередным числом нового интервала. Если длины входящих интервалов L то максимальная длина нового интервала 2*L. В случае же слияния предпоследнего интервала с остатком массива длина нового будет несколько меньше 2*L.

Переход к следующему элементу каждого из входящих интервалов выполняется только в том случае, если на текущем шаге элемент входящего интервала стал элементом нового интервала. Кроме того, перед переходом к следующему элементу выполняется проверка - достигнут или нет конец входящего интервала. Если конец достигнут, то переход не выполняется.