Stooge Sort (Придурковатая сортировка)

1. Общая постановка задачи

Необходимо создать новый компонент для калькулятора, который должен реализовывать новый интерфейс с методом сортировки Stooge sort (придурковатая сортировка). Интерфейс должен содержать методы с разными типами данных (int, long, float, double, long double).

2. Реализуемый алгоритм Stooge Sort

Stooge sort - это рекурсивный алгоритм сортировки, который имеет необычно высокую временную сложность O(n^2.7). Несмотря на свою неэффективность, алгоритм представляет интерес в учебных целях.

Алгоритм Stooge sort работает следующим образом:

Если первый элемент массива больше последнего, меняет их местами
Если в массиве более 2 элементов:
Рекурсивно сортирует первые 2/3 массива
Рекурсивно сортирует последние 2/3 массива
Снова рекурсивно сортирует первые 2/3 массива
Алгоритм получает свое неофициальное название "придурковатая сортировка" из-за своей неэффективности и странной логики работы.

3. Асимптотика

Временная сложность:
Средняя: O(n^2.7)
Лучшая: O(n^2.7)
Худшая: O(n^2.7)
Сложность по памяти: O(n) - из-за глубины рекурсии

Алгоритм Stooge sort является одним из самых неэффективных алгоритмов сортировки, но его реализация достаточно проста и наглядна для понимания рекурсивных алгоритмов.

4. Реализация

При решении задачи интерфейс содержит 5 методов для различных типов данных:

```
typedef struct IEcoLab1VTbl {

    /* IEcoUnknown */
    int16_t (ECOCALLMETHOD *QueryInterface)(/* in */ IEcoLab1Ptr_t me, /* in */ const UGUID* riid, /* out */ voidptr_t* ppv);
    uint32_t (ECOCALLMETHOD *AddRef)(/* in */ IEcoLab1Ptr_t me);
    uint32_t (ECOCALLMETHOD *Release)(/* in */ IEcoLab1Ptr_t me);

    /* IEcoLab1 */
    int16_t (ECOCALLMETHOD *stoogeSortInt)(/* in */ IEcoLab1Ptr_t me, /* in */ int* array, /* in */ int size);
    int16_t (ECOCALLMETHOD *stoogeSortLong)(/* in */ IEcoLab1Ptr_t me, /* in */ long* array, /* in */ int size);
    int16_t (ECOCALLMETHOD *stoogeSortFloat)(/* in */ IEcoLab1Ptr_t me, /* in */ float* array, /* in */ int size);
    int16_t (ECOCALLMETHOD *stoogeSortDouble)(/* in */ IEcoLab1Ptr_t me, /* in */ double* array, /* in */ int size);
    int16_t (ECOCALLMETHOD *stoogeSortLongDouble)(/* in */ IEcoLab1Ptr_t me, /* in */ long double* array, /* in */ int size);

} IEcoLab1VTbl, *IEcoLab1VTblPtr;
```

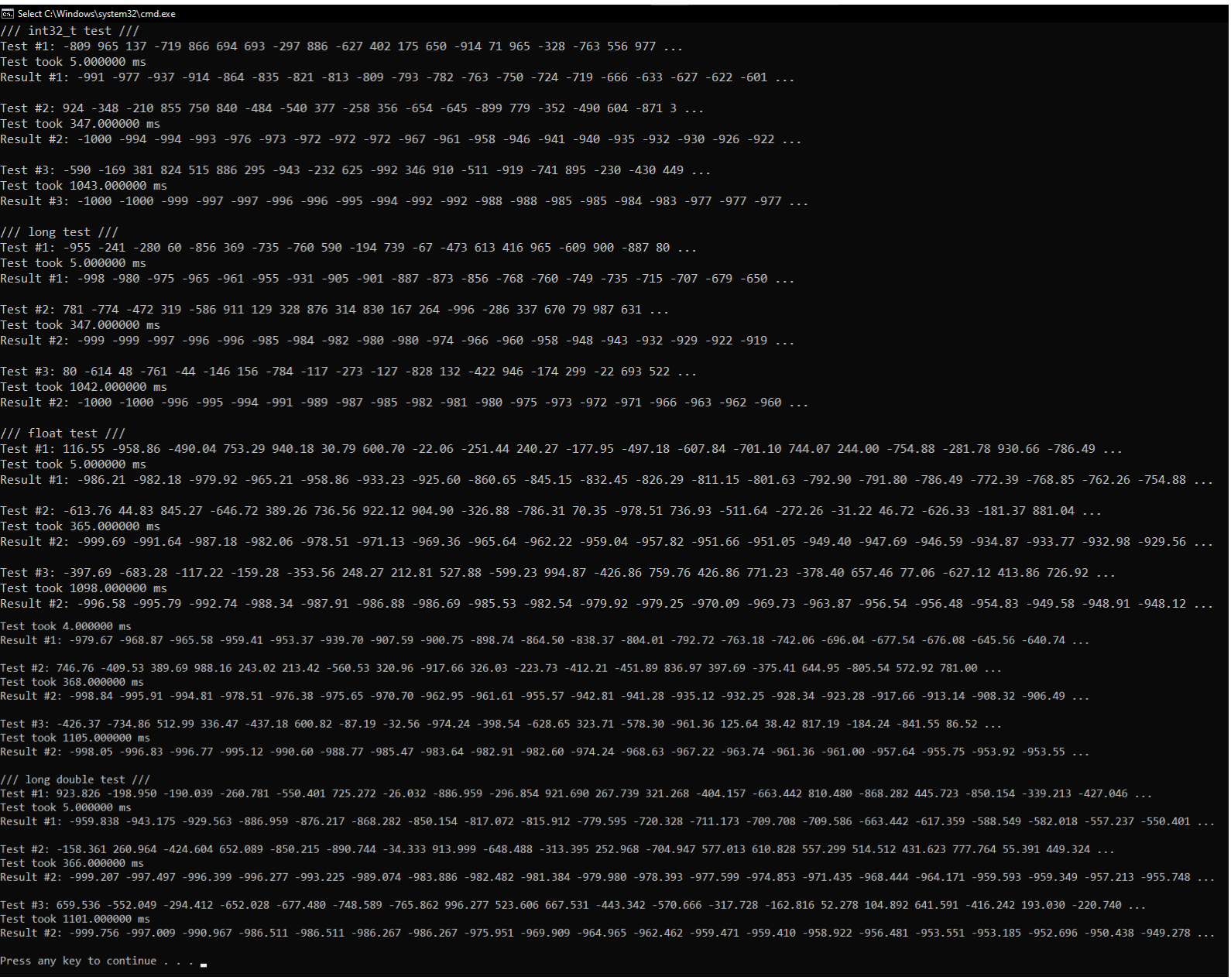
Каждый метод реализует рекурсивный алгоритм Stooge sort для соответствующего типа данных. Приведу пример основного и вспомогательного метода для целых чисел:

```
/*
 * Вспомогательные функции для Stooge sort
 */
static void stoogeSortHelperInt(int* arr, int l, int r) {
    if (arr[l] > arr[r]) {
        int temp = arr[l];
        arr[l] = arr[r];
        arr[r] = temp;
    }
    if (r - l + 1 > 2) {
        int t = (r - l + 1) / 3;
        stoogeSortHelperInt(arr, l, r - t);
        stoogeSortHelperInt(arr, l + t, r);
        stoogeSortHelperInt(arr, l, r - t);
    }
}
```

```
/*
 *
 * <сводка>
 *   Функция stoogeSortInt
 * </сводка>
 *
 * <описание>
 *   Функция сортирует массив целых чисел с помощью алгоритма Stooge sort
 * </описание>
 *
 */
static int16_t ECOCALLMETHOD CEcoLab1_stoogeSortInt(/* in */ IEcoLab1Ptr_t me, /* in */ int* array, /* in */ int size) {
    CEcoLab1* pCMe = (CEcoLab1*)me;
    /* Проверка указателей */
    if (me == 0 || array == 0 || size <= 0) {
        return ERR_ECO_POINTER;
    }
    stoogeSortHelperInt(array, 0, size - 1);

    return ERR_ECO_SUCCESES;
}
```

5. Пример работы



Программа работает с типами данных int, long, float, double, long double. Алгоритм корректно сортирует как положительные, так и отрицательные числа.

6. Сравнение с qsort из библиотеки stdlib.h

Результаты работы stoogeSortInt, stoogeSortLong, stoogeSortFloat, stoogeSortDouble и stoogeSortLongDouble в миллисекундах на 100, 500 и 1000 элементов с пятью типами данных:

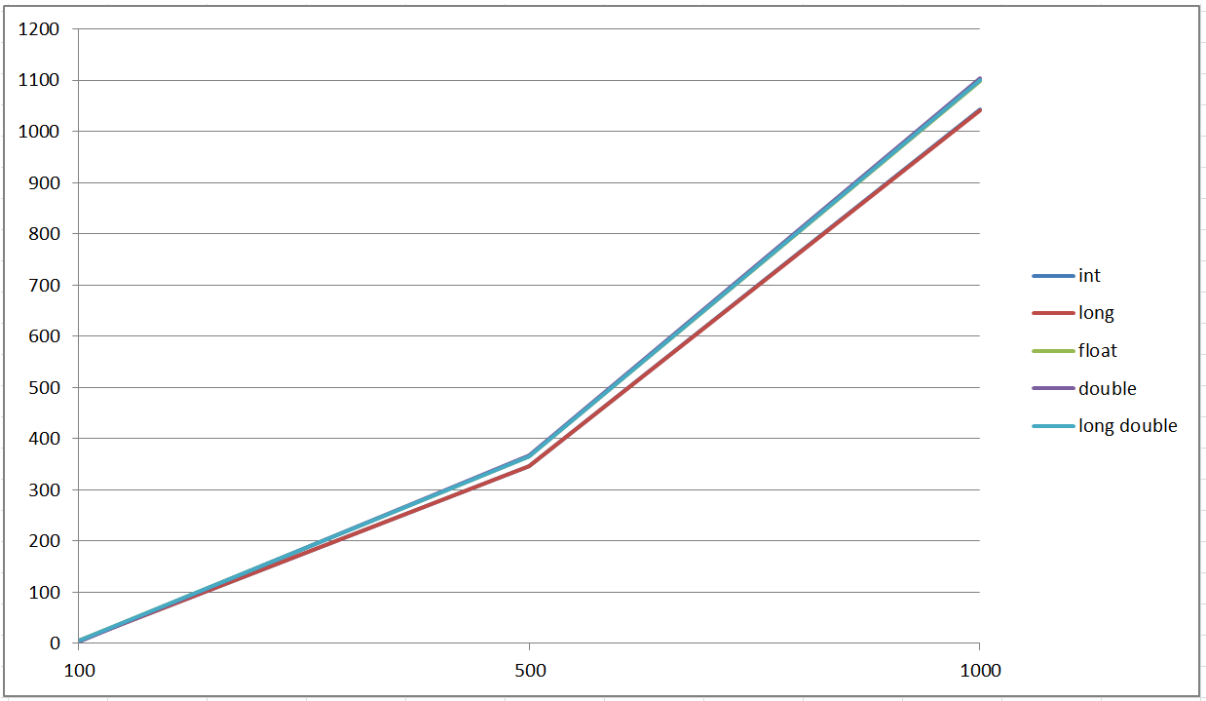
My StoogeSort					
	int	long	float	double	long double
100	5	5	5	4	5
500	347	347	365	368	366
1000	1043	1042	1098	1105	1101

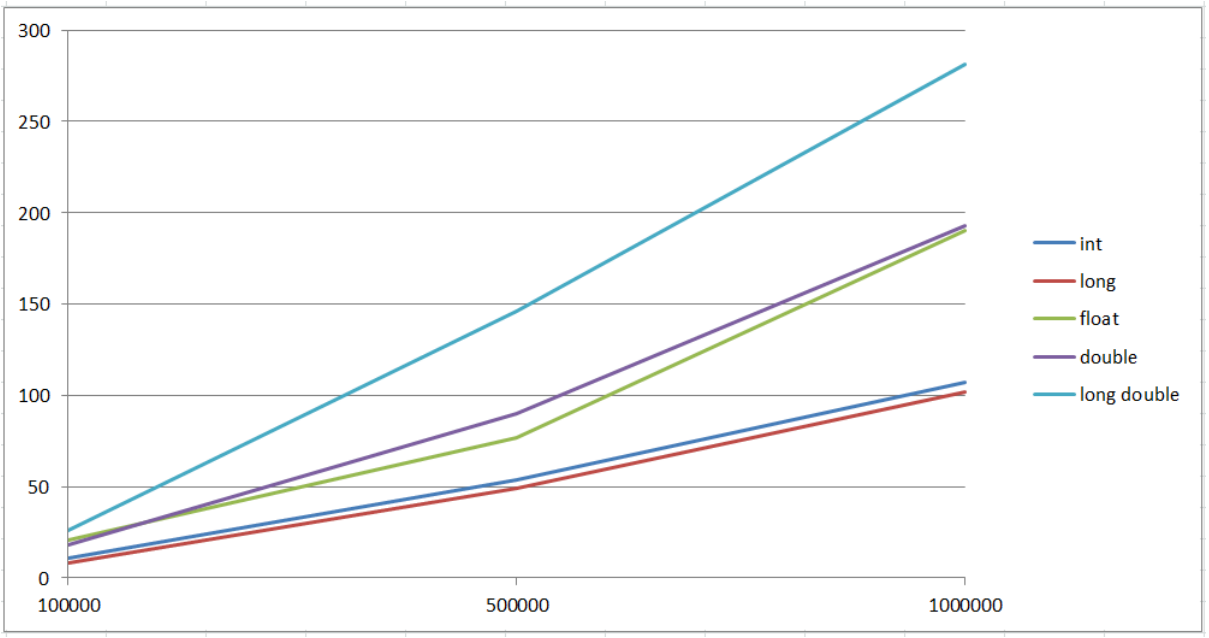
Результаты работы стандартной функции qsort из библиотеки stdlib.h в миллисекундах на 100000, 500000 и 1000000 элементов с пятью типами данных:

stdlib.h qsort					
	int	long	float	double	long double
100000	11	8	21	18	26
500000	54	49	77	90	146
1000000	107	102	190	193	281

Можно заметить, что алгоритм Stooge sort значительно уступает по производительности стандартной функции qsort, особенно при увеличении размера массива. Это ожидаемо, учитывая асимптотическую сложность O(n^2.7) против O(n log n) у qsort.

Также сравним диаграммы:





Разница в производительности становится критической при обработке массивов размером более 1000 элементов. Для массива из 1000 элементов Stooge sort работает примерно в 10 раз медленнее, чем qsort для массива из 1 000 000 элементов.

7. Выводы

Алгоритм Stooge sort представляет собой интересный учебный пример рекурсивного алгоритма, но его практическое применение ограничено из-за высокой временной сложности. Тем не менее, реализация этого алгоритма в рамках компонентной архитектуры Eco позволяет понять принципы работы с различными типами данных и демонстрирует важность выбора эффективного алгоритма для решения конкретной задачи.