

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный технологический университет «СТАНКИН»
ФГБОУ ВО МГТУ «СТАНКИН»

Кафедра «Управление и информатика в технических системах»
Дисциплина: «Алгоритмы и структуры данных»

Отчет по лабораторным работам

Выполнил: студент группы ИДБ-22-10

Наумов Д.А.

Преподаватель: доцент Евдокимов С.А.

Оглавление

Обменная сортировка со слиянием	2
Постановка задачи.....	2
Описание алгоритма.....	2
Графики времени сортировки	4
Выводы	5
Литература	5

Обменная сортировка со слиянием

Постановка задачи

Написать программу обменной сортировки со слиянием и сравнить ее время выполнения со стандартной функцией `qsort`. Для сравнения вычисляем время выполнения функции сортировки на массиве целых чисел следующих размеров: 16, 100, 500, 1000, 5000.

Для работы программы сортировки генерируем случайные массивы целых чисел:

- для массивов размерами ≤ 500 элементов числа в интервале $[100, 1000)$;
- для массивов размерами > 500 элементов числа в интервале $[1000, 10000)$;

Для проверки работы программы исходный не сортированный массив выводим в файл с именем d<число элементов>.txt, а сортированный массив выводим в файл с именем d<число элементов>-s.txt.

Время работы функции сортировки рассчитываем с точностью до тысячных долей миллисекунд. Например, 0.002, 0.018, 0.377, 1.1380, 35.806 мсек.

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов (N) в массиве:

- график функции N^2 ;
- график функции $N \log_2 N$;
- график времени работы стандартной функции `qsort`;
- график времени работы функции обменной сортировки со слиянием;

Графики создаем в табличном процессоре Microsoft Excel.

Описание алгоритма

Обменная сортировка со слиянием относится к классу обменных сортировок. Основной алгоритм состоит в следующем:

1. Имеется массив целых чисел (R) из N элементов от 0 до N - 1.
2. Необходимо найти t – наименьшее целое число, такое что $2^t \geq N$.
3. p – это 2^{t-1} – кол-во раз которое будут выполняться последующие шаги.
4. Далее алгоритм сортирует N элементов путем независимой сортировки подмассивов, а затем, на последней итерации, когда $p=1$, происходит слияние двух рассортированных последовательностей.

Формальная запись алгоритма посредством диаграмм Насси-Шнайдермана показана на рис.1.

Результат работы программы на массиве из 16 чисел показан ниже:

Командная строка:

Введите название файла:

d16.txt

Результат работы программы:

Сортируется массив из 16 чисел

293, 180, 389, 639, 928, 107, 324, 723, 422, 840, 206, 599, 445, 359, 841, 378

107, 180, 206, 293, 324, 359, 378, 389, 422, 445, 599, 639, 723, 840, 841, 928

Массив отсортирован

Время выполнения: 0.013 мсек

Сортированный массив записан в файл: d16-s.txt

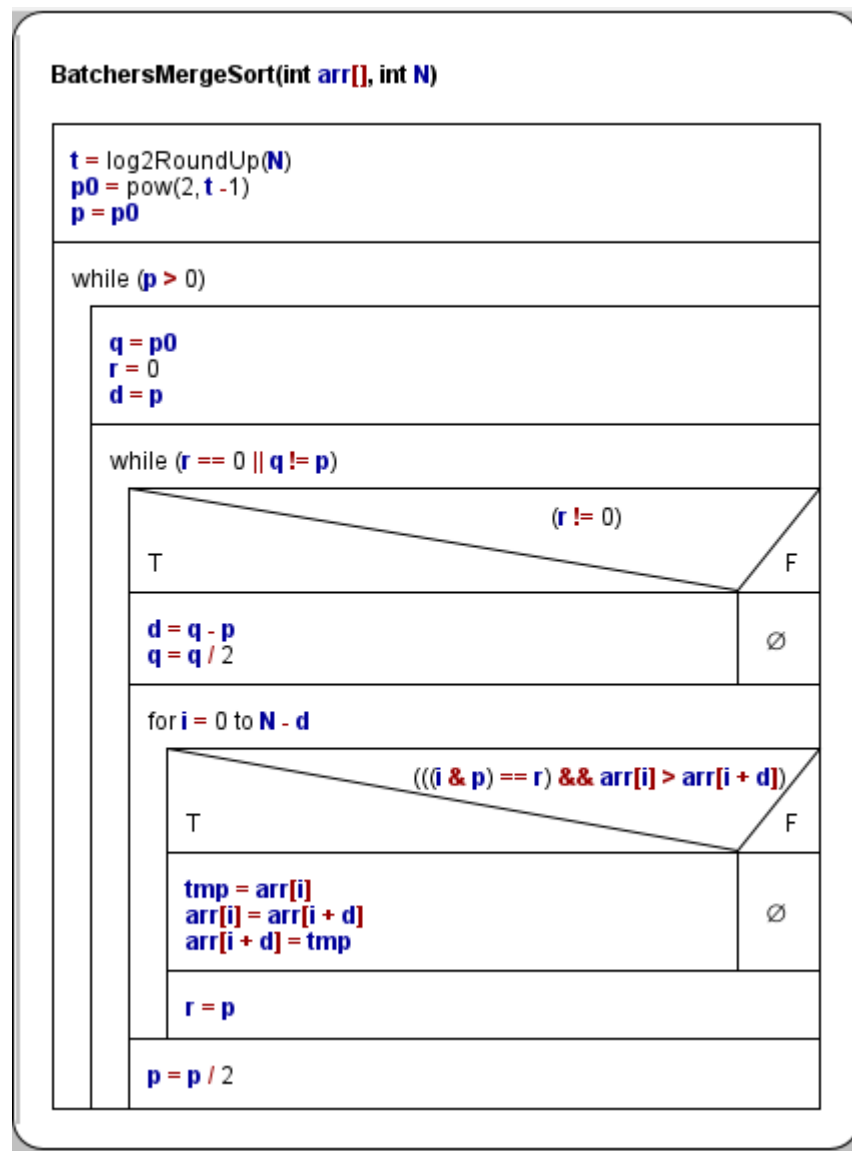


Рис.1. Запись алгоритма обменной сортировки со слиянием посредством диаграммы Насси - Шнайдермана

Исходный текст на C/C++ функции сортировки, составленный по алгоритму на рис.1 показан ниже:

```
void BatchersMergeSort(int arr[], int N) {
    int t = log2RoundUp(N);
    int p0 = pow(2, t - 1);
    int p = p0;

    while (p > 0) {
        int q = p0, r = 0, d = p;

        while (r == 0 || q != p) {
            if (r != 0)
            {
                d = q - p;
                q = q / 2;
            }

            for (int i = 0; i < N - d; i++) {
                if (((i & p) == r) && arr[i] > arr[i + d]) {
                    swap(arr[i], arr[i + d]);
                }
            }

            r = p;
        }

        p = p / 2;
    }
}
```

Функция log2RoundUp(N) подсчитывает значение t

Графики времени сортировки

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов (N) в массиве:

- график функции N^2 ;
- график функции $N \cdot \log_2 N$;
- график времени работы стандартной функции qsort;
- график времени работы функции BatchersMergeSort сортировки простыми вставками;
- график времени работы функции BubbleSort – сортировки пузырьком;

N	N^2	$N \cdot \log_2 N$	мксек.		
			qsort	BatchersMergeSort	BubbleSort
16	256	64	7	14	2
100	10000	664	16	29	73
500	250000	4483	81	146	1952
1000	1000000	9966	169	333	7843
5000	25000000	61439	983	2407	202262

Табл.1. Данные для создания графиков времени сортировки

Логарифмическая шкала по оси Y

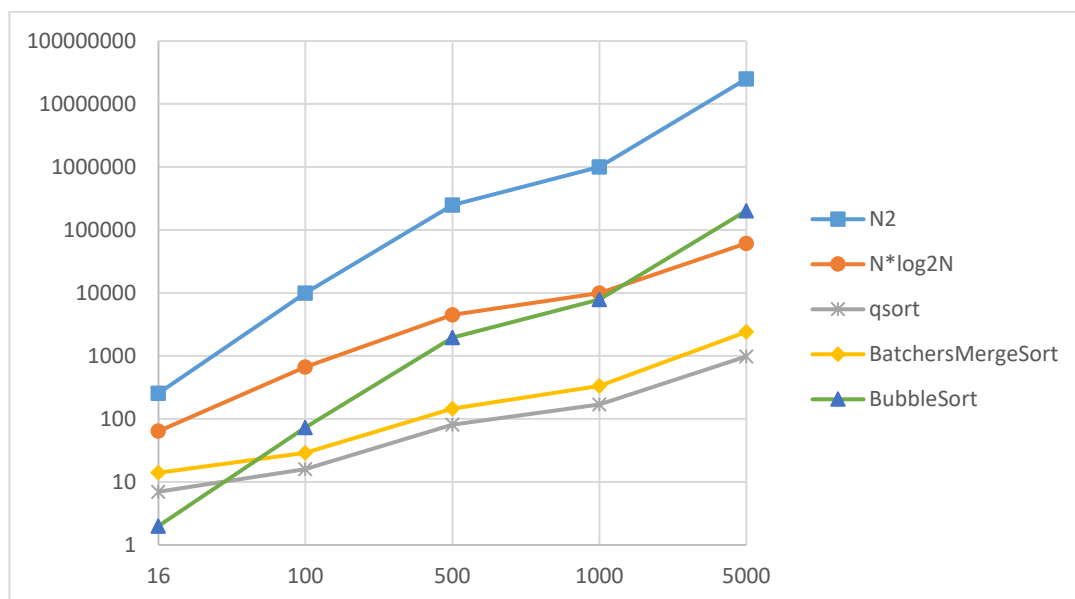


Рис.2. Графики времени сортировки различными алгоритмами по Табл.1.

Выводы

Из анализа графиков на рис.2 следует:

1. Обменная сортировка со слиянием имеет временную сложность $\frac{1}{2} [\log_2 N]([\log_2 N] + 1)$
2. Обменная сортировка со слиянием работает чуть медленнее, чем встроенная функция qsort, но стоит учитывать, что сортировка Бэтчера подразумевает возможность одновременного параллельного вычисления с нескольких компьютеров. Эта сортировка является одной из самых быстрых в мире.
3. На очень небольших данных (примерно меньше 50 чисел) уступает сортировке пузырьком.

Литература

1. Д.Кнут. Искусство программирования для ЭВМ. Сортировка и поиск. Том 3. М.:Мир, Москва, 1978. — 844 с.