

Занятие 4. Определение эффективного алгоритма сортировки

Цель. Получить навыки по анализу вычислительной сложности нескольких алгоритмов сортировки и определение наиболее эффективного алгоритма

Разработать три алгоритма сортировки, определенные вариантом. Провести анализ вычислительной и емкостной сложности алгоритма на массивах, заполненных случайно. Определить наиболее эффективный алгоритм.

Задание 1. Определение эффективного алгоритма в среднем случае

Требования по выполнению задания

1. Разработать алгоритм простой сортировки, определенной в варианте, реализовать алгоритм. Сформировать таблицу Таблица 2 результатов сортировки по формату Таблица 1 для массива, заполненного случайными числами. Определить емкостную сложность алгоритма.
Определить асимптотическую сложность алгоритма.
2. Разработать алгоритм ускоренной сортировки, определенной в варианте, реализовать алгоритм. Сформировать таблицу Таблица 3 результатов сортировки по формату Таблица 1 для массива, заполненного случайными числами. Определить емкостную сложность алгоритма.
Определить асимптотическую сложность алгоритма.
3. Выполнить анализ полученных результатов по таблицам 2 и 3. Определить эффективный из алгоритмов по временной сложности.
4. Представить график зависимости $C_f + M_f$ для анализируемых алгоритмов.
5. Разработать алгоритм ускоренной сортировки «Прямое слияние», реализовать алгоритм. Сформировать таблицу Таблица 4 результатов сортировки по формату Таблица 1 для массива, заполненного случайными числами. Определить емкостную сложность алгоритма.
Определить асимптотическую сложность алгоритма.
6. Выполнить анализ полученных результатов по таблицам 3 и 4. Определить эффективный из алгоритмов по временной сложности.
7. Представить график зависимости $C_f + M_f$ для анализируемых алгоритмов.

Таблица 1. Сводная таблица результатов

n	T	f(C+M)	C_f+M_f
100			
1000			
10000			
100000			
1000000			

Таблица 1. Варианты индивидуальных заданий

Вариант	Алгоритм простой сортировки	Алгоритм усовершенствованной сортировки	Алгоритм слияния
1	Простого обмена (пузырек)	Шейкерная	Простое слияние
2	Простого обмена (пузырек) с условием Айверсона	Шейкерная сортировка	Простое слияние
3	Простого обмена (пузырек) с условием Айверсона	Шейкерная с условием Айверсона	Простое слияние
4	Простой вставки	Сортировка Шелла со смещениями Д. Кнута. Способ 1	Простое слияние
5	Простой вставки	Шелла со смещениями Д. Кнута. Способ 2	Простое слияние
6	Простой вставки	Шелла со смещениями Р. Седжвика.	Простое слияние
7	Простого выбора	Пирамидальная сортировка	Простое слияние
8	Простого выбора	Турнирная сортировка	Простое слияние

Методы определения смещения для сортировки Шелла, предложенные Д.Кнут и Р.Седжвик

Перед выполнением сортировки происходит вычисление длин промежутков (значения d из примера сортировки Шелла), которые записываются в массив, например, d .

По Седжвику

Значение смещения, записываемого в элемент массива d_i вычисляется по формуле:

$$k[i] = \begin{cases} 9 * 2^i - 9 * 2^{i/2} + 1 & \text{при } i - \text{четном} \\ 8 * 2^i - 6 * 2^{(i+1)/2} + 1 & \text{при } i - \text{нечетном} \end{cases}$$

Остановить создание и заполнение массива d на значении $d[i-1]$, если $3 * d[i] > n$ (размера массива).

По Кнуту

Определение длины промежутков методом предложенным Кнутом метод:

Способ 1:

$$t = \log_3 n - 1 \quad d_t = 1, \quad d[i-1] = 3 * d[i] + 1 \text{ т.е. } 1, 4, 13, 40, 121, \dots$$

Способ 2:

$$t = \log_2 n - 1, d_t = 1, d[i-1] = 2 * d[i] + 1 \text{ т.е. } 1, 3, 7, 13, 31, \dots$$

Задание 2. Определение эффективного из алгоритмов для наихудшего и наилучшего случаев

Требования по выполнению задания

1. Провести дополнительные прогоны программы на рабочих массивах, отсортированных строго в убывающем и возрастающем порядке значений элементов. Заполнить таблицы для каждого алгоритма по формату Таблица 1.
2. Провести анализ зависимости (или независимости) алгоритмов усовершенствованных сортировок сортировки от исходной упорядоченности массива на основе результатов, представленных в таблицах.
3. Определить асимптотическую сложность алгоритма в наихудшем и наилучшем случаях.
4. Определить эффективный в лучшем и худшем случае алгоритм.
5. Определить наиболее эффективный алгоритм для всех трех случаев.

Форма отчета

Отчет по заданию 1

Представить отчет по выполнению каждого пункта задания. Последовательность изложения в отчете в соответствии с порядком требований.

1. По задачам п.1, п.2, п.3 составить отчет в соответствии с требованиями оформления отчета по разработке программы (постановка задачи, описание подхода к решению, алгоритм, определение функции зависимости временной сложности алгоритма от размера массива и определение асимптотической сложности, код функции сортировки, тесты).
2. Отчеты по остальным пунктам выполнить в соответствии с их задачами.

Отчет по заданию 2

1. Представить таблицы с результатами прогонов на упорядоченных массивах.
2. Представить асимптотическую вычислительную сложность для каждого алгоритма для лучшего и худшего случаев. Привести ваши выводы по эффективному алгоритму.
3. Заполнить таблицу для рассматриваемых в задании алгоритмов

Алгоритм	Асимптотическая сложность алгоритма			
	Наихудший случай	Наилучший случай	Средний случай	Емкостная сложность

