**Практическое задание 8**

**Тема.** Эффективные алгоритмы сортировки.

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

**Задание.**

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

1. Разработать алгоритм простой сортировки, определенной вариантом. Определить емкостную и временную сложность алгоритма. Провести контрольные прогоны алгоритма для трех случаев входного массива разных размеров. Сформировать таблицы результатов №1-3 сортировки для массива, заполненного случайными числами, отсортированного и отсортированного в обратном порядке соответственно.
2. Разработать алгоритм усовершенствованной сортировки, определенной вариантом. Определить емкостную и временную сложность алгоритма. Провести контрольные прогоны алгоритма для трех случаев входного массива разных размеров. Сформировать таблицы результатов №4-6 сортировки для массива, заполненного случайными числами, отсортированного и отсортированного в обратном порядке соответственно.
3. Разработать алгоритм сортировки методом простого слияния. Определить емкостную и временную сложность алгоритма. Провести контрольные прогоны алгоритма для трех случаев входного массива разных размеров. Сформировать таблицы результатов №7-9 сортировки для массива, заполненного случайными числами, отсортированного и отсортированного в обратном порядке соответственно.
4. Выполнить анализ полученных результатов по составленным таблицам №1, №4 и №7. Определить наиболее эффективный из алгоритмов.
5. Представить график зависимости С+М для анализируемых алгоритмов.
6. Провести анализ зависимости (или независимости) алгоритмов сортировок от исходной упорядоченности массива на основе результатов, представленных в таблицах №2-3, №5-6 и №8-9.
7. Определить эффективный в лучшем и худшем случае алгоритм.

Таблица 1. Пример сводной таблицы результатов тестирования программы

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **T** | **f(n)=N2** | **C+M** |
| 100 | 0.01 | 10000 | 76543 |
| 1000 | 0.097 | 1000000 | 7645207 |
| 10000 | 0.8 | 100000000 | 764432855 |
| 100000 | 6 | 10000000000 | 76440354874 |
| 1000000 | 40 | 1000000000000 | 7644104878632 |

Таблица 2. Варианты заданий

|  |  |  |  |
| --- | --- | --- | --- |
| **Вариант** | **Алгоритм простой сортировки** | **Алгоритм усовершенствованной сортировки** | **Алгоритм**  **слияния** |
| 1 | Простого обмена (пузырек) | Шейкерная | Простое слияние |
| 2 | Простого обмена (пузырек) с условием Айверсона | Хоара | Простое слияние |
| 3 | Простого обмена (пузырек) с условием Айверсона | Шейкерная с условием Айверсона | Простое слияние |
| 4 | Простой вставки | 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 |

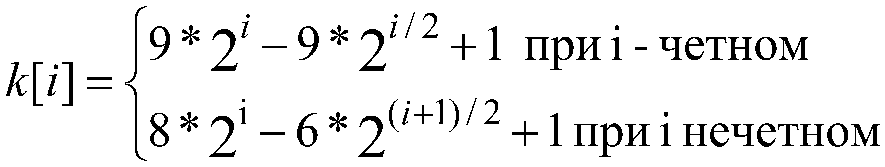
**Примечания.**

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

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? вычисляется по формуле:



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.

Способ 1:

t=log**3**n-1 dt =1, d[i-1]=3\*d[i]+1 т.е. 1, 4, 13, 40, 121, …..

Способ 2:

t=log2n-1 dt =1, d[i-1]=2\*d[i]+1 т.е. 1, 3, 7, 13, 31, …..