

## ПРАКТИЧЕСКАЯ РАБОТА № 3

### ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ

**Цель работы** – познакомиться с различными алгоритмами сортировки и поиска, оценить их эффективность по памяти и количеству операций.

#### Постановка задачи

Написать две программы согласно номеру индивидуального варианта.

В **первой программе** провести сравнение указанных алгоритмов сортировки массивов, содержащих  $N1$ ,  $N2$ ,  $N3$  и  $N4$  элементов. Каждую функцию сортировки вызывать трижды: для сортировки неупорядоченного массива, упорядоченного массива и массива, упорядоченного в обратном порядке. При работе каждого алгоритма сортировки выполнить подсчет основных (производимых над элементами массива) и вспомогательных (всех остальных) операций, указанных в вариативной части задания (сравнений или присваиваний), а также зафиксировать время работы алгоритма. Сортируемая последовательность для всех методов должна быть одинаковой (считывать необходимое количество элементов из прилагаемого файла *test\_numbers.txt*). Оценить время работы и эффективность алгоритмов сортировки по заданному критерию и объему требуемой дополнительно памяти.

При выполнении задания на повышенном уровне сложности дополнительно провести анализ того, как наличие повторяющихся ключей во входной последовательности влияет на трудоемкость каждого из рассматриваемых алгоритмов сортировки. Для этого создать четыре файла, содержащих  $N4$  неупорядоченных чисел, в которых значения элементов будут повторяться по 10, 100, 500 и 1000 раз, и так же, как в первой половине работы, выполнить сортировку этих последовательностей (неупорядоченных, упорядоченных и упорядоченных в обратном порядке) каждым из методов.

Во **второй программе** реализовать две указанные структуры данных, заполнив их значениями из приложенного файла *test\_numbers.txt*. Выполнить поиск 100 ключей в указанных структурах данных, для каждого ключа выводить сообщение о том, найден он или нет, и количество выполненных при поиске сравнений ключей, в конце программы вывести среднее количество сравнений, пришедшееся на один ключ. При формировании тестового набора включать в него как имеющиеся в файле, так и отсутствующие в нем ключи (меньшие 100000000, большие 1000000000 и принадлежащие интервалу  $[100000000; 1000000000)$ ). Оценить количество требуемой памяти для реализации каждой структуры и количество сравнений при поиске.

При оформлении **отчета** перед каждой программой привести теоретические расчеты трудоемкости по заданному критерию в нотациях  $O$  и  $\Omega$ . После результатов работы программ сделать выводы о правильности выполненных расчетов и выбрать из анализируемых наиболее

эффективный алгоритм. После первой программы, помимо этого, вычислить количество используемой каждым алгоритмом дополнительной памяти, а после второй – количество памяти, используемой каждой реализованной структурой данных. По результатам проведенных вычислений сделать выводы о соотношении трудоемкости и объема дополнительной памяти для анализируемых алгоритмов.

### **Варианты заданий**

#### ***Вариант № 1***

1. Порядок: по убыванию элементов. Методы: выбора, пузырька, пирамидальная сортировка, быстрая сортировка.  $N1=5000$ ,  $N2=30000$ ,  $N3=100000$ ,  $N4=120000$ . Критерий – количество сравнений.

2. AVL-дерево, бор. Подсчет выделяемой памяти произвести в программе.

#### ***Вариант № 2***

1. Порядок: по возрастанию элементов. Методы: выбора, пузырька, пирамидальная сортировка, быстрая сортировка.  $N1=10000$ ,  $N2=30000$ ,  $N3=70000$ ,  $N4=100000$ . Критерий – количество присваиваний.

2. Несбалансированное бинарное дерево поиска, декартово дерево.

#### ***Вариант № 3***

1. Порядок: по возрастанию элементов. Методы: пузырька, шейкера, быстрая сортировка, сортировка естественным слиянием.  $N1=10000$ ,  $N2=50000$ ,  $N3=100000$ ,  $N4=150000$ . Критерий – количество сравнений.

2. Splay-дерево, AVL-дерево. При формировании тестового набора ключей учесть, что вероятности поиска разных ключей должны различаться.

#### ***Вариант № 4***

1. Порядок: по убыванию элементов. Методы: пузырька, шейкера, быстрая сортировка, сортировка естественным слиянием.  $N1=20000$ ,  $N2=50000$ ,  $N3=80000$ ,  $N4=110000$ . Критерий – количество присваиваний.

2. AVL-дерево, хеш-таблица с разрешением коллизий методом цепочек (хеш-функция – метод деления).

#### ***Вариант № 5***

1. Порядок: по убыванию элементов. Методы: пузырька, пузырька с фиксацией места обмена, шейкера, быстрая сортировка.  $N1=10000$ ,  $N2=18000$ ,  $N3=30000$ ,  $N4=60000$ . Критерий – количество сравнений.

2. Упорядоченный односвязный линейный список, красно-черное дерево.

### **Вариант № 6**

1. Порядок: по возрастанию элементов. Методы: выбора, простых вставок, Шелла (шаг сортировки  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), пирамидальная сортировка.  $N1=50000$ ,  $N2=90000$ ,  $N3=120000$ ,  $N4=150000$ . Критерий – количество присваиваний.

2. Красно-черное дерево, декартово дерево.

### **Вариант № 7**

1. Порядок: по возрастанию элементов. Методы: простых вставок, бинарных вставок, Шелла (шаг сортировки  $h_{k-1}=3h_k+1$ ,  $h_t=1$ ,  $t=\log_3 n-1$  и  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ).  $N1=10000$ ,  $N2=30000$ ,  $N3=70000$ ,  $N4=100000$ . Критерий – количество сравнений.

2. Упорядоченный двусвязный линейный список, хеш-таблица с разрешением коллизий методом открытой адресации (хеш-функция – метод свертки).

### **Вариант № 8**

1. Порядок: по убыванию элементов. Методы: пирамидальная сортировка, метод Хоара, сортировка прямым слиянием, сортировка естественным слиянием.  $N1=5000$ ,  $N2=100000$ ,  $N3=150000$ ,  $N4=200000$ . Критерий – количество сравнений.

2. Несбалансированное дерево бинарного поиска, splay-дерево. При формировании тестового набора ключей учесть, что вероятности поиска разных ключей должны различаться.

### **Вариант № 9**

1. Порядок: по убыванию элементов. Методы: быстрая сортировка, сортировка прямым слиянием, сортировка естественным слиянием, поразрядная сортировка.  $N1=10000$ ,  $N2=60000$ ,  $N3=110000$ ,  $N4=160000$ . Критерий – количество присваиваний.

2. Бор, хеш-таблица с разрешением коллизий методом цепочек (хеш-функция – метод деления).

### **Вариант № 10**

1. Порядок: по возрастанию элементов. Методы: пирамидальная сортировка, метод Хоара, сортировка прямым слиянием, сортировка естественным слиянием.  $N1=50000$ ,  $N2=90000$ ,  $N3=120000$ ,  $N4=150000$ . Критерий – количество присваиваний.

2. Декартово дерево, AVL-дерево.

### **Вариант № 11**

1. Порядок: по убыванию элементов. Методы: выбора, простых вставок, Шелла (шаг сортировки  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), пирамидальная сортировка.  $N1=15000$ ,  $N2=40000$ ,  $N3=80000$ ,  $N4=120000$ . Критерий – количество сравнений.

2. AVL-дерево, упорядоченный односвязный линейный список.

### **Вариант № 12**

1. Порядок: по возрастанию элементов. Методы: выбора, простых вставок, Шелла (шаг

сортировки  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), пирамидальная сортировка.  $N1=10000$ ,  $N2=30000$ ,  $N3=70000$ ,  $N4=100000$ . Критерий – количество присваиваний.

2. Несбалансированное бинарное дерево поиска, бор.

#### **Вариант № 13**

1. Порядок: по убыванию элементов. Методы: пузырька, пузырька с фиксацией факта обмена, пузырька с фиксацией места обмена, шейкера.  $N1=5000$ ,  $N2=15000$ ,  $N3=30000$ ,  $N4=50000$ . Критерий – количество сравнений.

2. Красно-черное дерево, AVL-дерево.

#### **Вариант № 14**

1. Порядок: по возрастанию элементов. Методы: простых вставок, бинарных вставок, Шелла (шаг сортировки  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), Шелла (шаг сортировки задается числами Фибоначчи).  $N1=5000$ ,  $N2=15000$ ,  $N3=45000$ ,  $N4=135000$ . Критерий – количество присваиваний.

2. Красно-черное дерево, хеш-таблица с разрешением коллизий методом цепочек (хеш-функция – метод деления).

#### **Вариант № 15**

1. Порядок: по убыванию элементов. Методы: выбора, простых вставок, Шелла (шаг сортировки  $h_{k-1}=3h_k+1$ ,  $h_t=1$ ,  $t=\log_3 n-1$ ), сортировка прямым слиянием.  $N1=10000$ ,  $N2=18000$ ,  $N3=30000$ ,  $N4=60000$ . Критерий – количество сравнений.

2. Декартово дерево, бор. Подсчет выделяемой памяти произвести в программе.

#### **Вариант № 16**

1. Порядок: по убыванию элементов. Методы: шейкера, Шелла (шаг сортировки  $h_{k-1}=3h_k+1$ ,  $h_t=1$ ,  $t=\log_3 n-1$ ), сортировка прямым слиянием, поразрядная сортировка.  $N1=10000$ ,  $N2=50000$ ,  $N3=100000$ ,  $N4=150000$ . Критерий – количество присваиваний.

2. Красно-черное дерево, бор. Подсчет выделяемой памяти произвести в программе.

#### **Вариант № 17**

1. Порядок: по возрастанию элементов. Методы: пузырька, Шелла (шаг сортировки  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), Шелла (шаг сортировки задается числами Фибоначчи), поразрядная сортировка.  $N1=10000$ ,  $N2=30000$ ,  $N3=70000$ ,  $N4=100000$ . Критерий – количество присваиваний.

2. Упорядоченный двусвязный список, несбалансированное дерево бинарного поиска.

#### **Вариант № 18**

1. Порядок: по убыванию элементов. Методы: бинарных вставок, Шелла (шаг сортировки  $h_{k-1}=3h_k+1$ ,  $h_t=1$ ,  $t=\log_3 n-1$  и  $h_{k-1}=2h_k+1$ ,  $h_t=1$ ,  $t=\log_2 n-1$ ), сортировка естественным слиянием.  $N1=15000$ ,  $N2=40000$ ,  $N3=80000$ ,  $N4=150000$ . Критерий – количество сравнений.

2. Несбалансированное бинарное дерево поиска, красно-черное дерево.

#### **Вариант № 19**

1. Порядок: по убыванию элементов. Методы: выбора, пузырька с фиксацией места обмена, бинарных вставок, поразрядная.  $N_1=10000$ ,  $N_2=30000$ ,  $N_3=50000$ ,  $N_4=90000$ . Критерий – количество присваиваний.

2. Хеш-таблица с разрешением коллизий методом цепочек, хеш-таблица с разрешением коллизий методом открытой адресации (хеш-функция – метод деления).

### ***Вариант № 20***

1. Порядок: по возрастанию элементов. Методы: шейкера, сортировка прямым слиянием, сортировка естественным слиянием, поразрядная сортировка.  $N_1=50000$ ,  $N_2=90000$ ,  $N_3=120000$ ,  $N_4=150000$ . Критерий – количество присваиваний.

2. Хеш-таблица с разрешением коллизий методом цепочек (хеш-функция – метод свертки), splay-дерево. При формировании тестового набора ключей учесть, что вероятности поиска разных ключей должны различаться.

### **Контрольные вопросы**

1. Что такое сортировка?
2. Какие виды сортировки Вы знаете?
3. Какие методы сортировки называются устойчивыми?
4. Какое среднее количество сравнений требуется для прямых и улучшенных методов сортировки?
5. На какие категории можно разбить прямые методы сортировки?
6. Опишите алгоритм сортировки методом пузырька
7. Опишите алгоритм сортировки методом пузырька с фиксацией факта обмена
8. Опишите алгоритм сортировки методом вставки
9. Опишите алгоритм сортировки методом прямого выбора
10. Опишите алгоритм сортировки методом пузырька с фиксацией места обмена
11. Опишите алгоритм шейкерная сортировки
12. Опишите алгоритм сортировки методом Шелла
13. Опишите алгоритм сортировки с помощью пирамиды
14. Опишите алгоритм сортировки Хоара
15. Опишите алгоритм поразрядной сортировки
16. Какие методы сортировки файлов Вы знаете?
17. Какие особенности требуется учитывать при сортировке файлов?
18. Опишите алгоритм сортировки методом прямого слияния
19. Опишите алгоритм сортировки методом естественного слияния
20. Опишите алгоритм сортировки методом многопутевого слияния
21. Опишите алгоритм многофазной сортировки
22. Какой метод сортировки обычно используется для подготовки серий перед сортировкой

файлов и почему выбран именно он?

23. Что такое поиск? Какие виды поиска Вы знаете
24. Что такое ключ поиска? Какие бывают ключи поиска?
25. Методы поиска в неупорядоченной таблице
26. Методы поиска в упорядоченной последовательности
27. Что такое поисковый индекс?
28. Что такое индекс в индексно-последовательном поиске?
29. Что такое дерево бинарного поиска?
30. Как добавить элемент в дерево бинарного поиска?
31. Как удалить элемент из дерева бинарного поиска?
32. Какие виды сбалансированных деревьев бинарного поиска Вы знаете?
33. Методы балансировки бинарного дерева.
34. Как добавить элемент в красно-черное дерево?
35. Как удалить элемент из красно-черного дерева?
36. Как добавить элемент в декартово дерево?
37. Как удалить элемент из декартова дерева?
38. Что такое «метод ножниц и клея»? Для каких деревьев он применяется?
39. Деревья цифрового поиска. Бор
40. Как добавить элемент в бор?
41. Как удалить элемент из бора?
42. Б-деревья
43. Поиск с использованием хеш-таблиц.
44. Разрешение коллизий при хешировании методом открытой адресации
45. Разрешение коллизий при хешировании методом цепочек