

1. Цель работы

Целью работы является изучение алгоритмов внутренней сортировки и получение практических навыков их использования, и анализа их сложности.

2. Задание на лабораторную работу

Использовать неупорядоченный массив A, содержащий n целочисленных элементов. Величина n определяется по согласованию с преподавателем. Дополнительно в программе должны быть реализованы следующие функции:

- 1) Поиск элемента либо по его порядковой позиции, либо по его содержимому;
- 2) Добавление/удаление элемента с последующей пересортировкой последовательности;
- 3) В программе должен быть реализован подсчет количества сравнений и перестановок, при осуществлении сортировки – в сортировке подсчетом сравнения и перестановки не проводятся.

Вариант №1:

№ вар.	Задание	Алгоритм сортировки
1	Найти количество различных чисел среди элементов массива	Подсчетом

3. Описание созданных функций

Для реализации задания была использована структура Array:

Имя: void gen.

Назначение: генерация массива.

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

Имя: print.

Назначение: вывод массива в консоль.

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

Имя: sort.

Назначение: сортировка массива (подсчётом).

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

Имя: different.

Назначение: поиск различных чисел в массиве.

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

Имя: find.

Назначение: поиск значения.

Входные данные:

- Value – переменная хранящая в себе значение, которое требуется найти

Выходные данные:

- Индекс искомого значения

Побочный эффект: отсутствует.

Имя: append.

Назначение: добавление элемента в массив с последующей сортировкой.

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

Имя: remove.

Назначение: удаление элемента по индексу.

Входные данные:

- Index – переменная, хранящая в себе индекс заданный пользователем

Выходные данные:

Побочный эффект: отсутствует.

Имя: remove_by_value.

Назначение: удаление элемента по значению.

Входные данные:

- value – переменная, хранящая в себе удаляемое значение;

Выходные данные:

Побочный эффект: отсутствует.

Имя: menu.

Назначение: вывод в консоль пунктов меню.

Входные данные:

Выходные данные:

Побочный эффект: отсутствует.

4. Листинг программы

```
1. #include <iostream>
2. #include <time.h>
3. #include <algorithm>

4. using namespace std;

5. const int MAX_SIZE = 1000;
6. const int MAX_VALUE = 10; // A must contain only numbers in interval [0 ..
   MAX_VALUE)

7. // Array.hpp
8. class Array
9. {
10. int* A;
11. int N;

12. public:

13. Array(int n)
14. {
15. // выделение памяти
16. A = new int[MAX_SIZE];
17. N = n;
18. }

19. void gen();
20. void print();
21. void sort();
22. int different();
23. int find(int value);
24. void append(int value);
25. void remove(int index);
26. void remove_by_value(int value);

27. ~Array()
28. {
29. delete[] A;
30. }
31. };

32. // Array.cpp
33. void Array::gen()
34. {
35. for (int i = 0; i < N; i++)
36. {
37. A[i] = rand() % MAX_VALUE;
38. }
39. }

40. void Array::print()
41. {
42. for (int i = 0; i < N; i++)
43. {
44. cout << A[i] << " ";
45. }
46. cout << endl;
47. }

48. void Array::sort()
49. {
```

```

50. int B[MAX_VALUE];
51. for (int i = 0; i < MAX_VALUE; i++)
52. B[i] = 0;

53. for (int i = 0; i < N; i++)
54. {
55. B[A[i]]++;
56. }

57. int j = 0;
58. for (int i = 0; i < MAX_VALUE; i++)
59. {
60. for (int k = 0; k < B[i]; k++)
61. {
62. A[j++] = i;
63. }
64. }
65. }

66. int Array::different()
67. {
68. // Suppose A is sorted in increase order
69. int count = 1;
70. for (int i = 0; i < N - 1; i++)
71. {
72. if (A[i] != A[i + 1])
73. ++count;
74. }
75. return count;
76. }

77. // return index of element=value if found,
78. // -1 if not found
79. int Array::find(int value)
80. {
81. for (int i = 0; i < N; i++)
82. if (A[i] == value)
83. return i;
84. return -1;
85. }

86. void Array::append(int value)
87. {
88. A[N] = value; // N=6 [0 1 2 3 4 5] value=6, N = 7
89. ++N;
90. sort();
91. }

92. // remove element by index
93. void Array::remove(int index)
94. {
95. if (index < 0 || index >= N)
96. {
97. cout << "Incorrect index" << endl;
98. return;
99. }
100.     for (int i = index; i < N; i++)
101.     {
102.         A[i] = A[i + 1];
103.     }
104.     --N;
105. }

106.     // remove by value all elements
107. void Array::remove_by_value(int value)

```

```

108.     {
109.     while (1)
110.     {
111.     int index = find(value);
112.     if (index == -1) return;
113.     remove(index);
114.     }
115.     }

116.     void menu()
117.     {
118.     cout << "1.Генерация массива." << endl;
119.     cout << "2.Печать массива " << endl;
120.     cout << "3.Сортировка массива подсчётом " << endl;
121.     cout << "4.Поиск количества различных чисел среди элементов массива " <<
endl;
122.     cout << "5.Поиск индекса искомого значения в массиве " << endl;
123.     cout << "6.Добавление элемента в массив " << endl;
124.     cout << "7.Удаление элемента по индексу " << endl;
125.     cout << "8.Удаление элемента по значению " << endl;
126.     cout << "9.Выход из программы " << endl;

127.     }

128.     int main()
129.     {
130.     setlocale(LC_ALL, "Russian");
131.     srand(time(NULL));
132.     int choose = 0;
133.     int N,Value,Index;
134.     cout << "Input array size: " << endl;
135.     cin >> N;
136.     if (N > MAX_SIZE)
137.     {
138.     cout << "N is too big!";
139.     exit(1);
140.     }
141.     Array A(N);

142.     do
143.     {
144.     cout << "Выберите пункт меню: " << endl;
145.     menu();
146.     cin >> choose;

147.     switch (choose)
148.     {
149.     case 1:
150.     {
151.     A.gen();
152.     break;
153.     }
154.     case 2:
155.     {
156.     A.print();
157.     break;
158.     }
159.     case 3:
160.     {
161.     A.sort();
162.     break;
163.     }
164.     case 4:
165.     {
166.     int c = A.different();

```

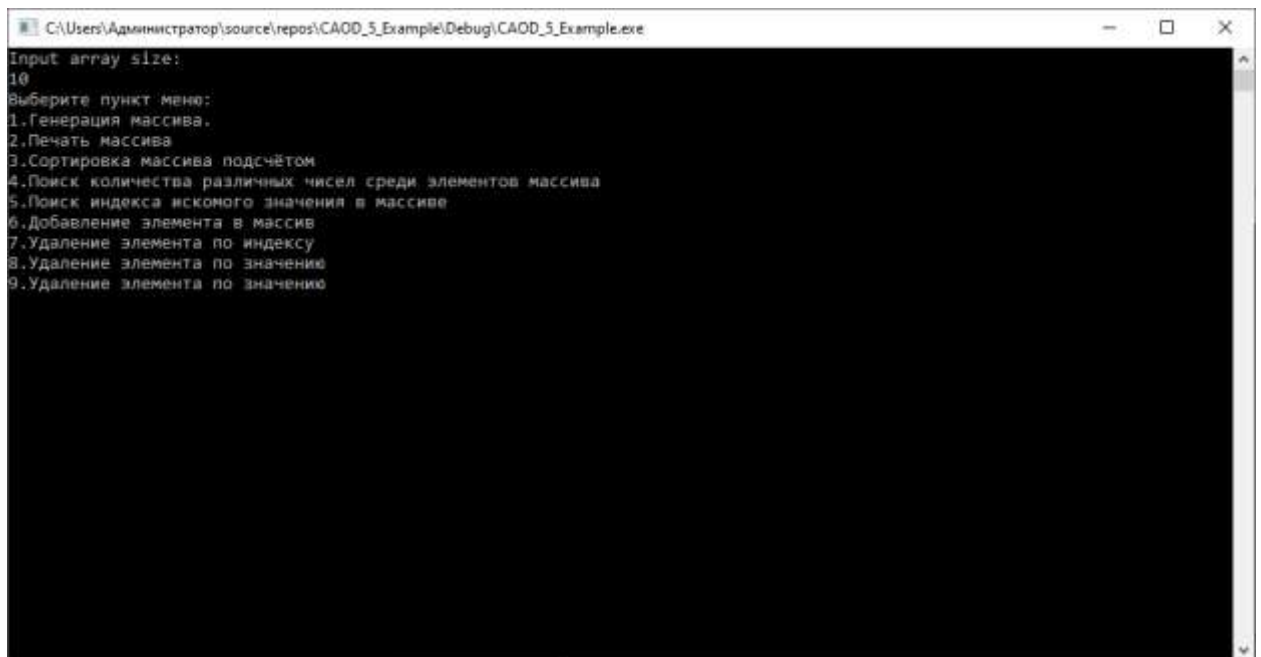
```
167.     cout << "Уникальных элементов в массиве: " << c << endl;
168.     break;
169. }
170. case 5:
171. {
172.     cout << "Insert find value: ";
173.     cin >> Value;
174.     int c = A.find(Value);
175.     cout << "Индекс искомого элемента равен: " << c << endl;
176.     break;
177. }
178. case 6:
179. {
180.     cout << "Insert add value: ";
181.     cin >> Value;
182.     A.append(Value);
183.     break;
184. }
185. case 7:
186. {
187.     cout << "Insert index remove value: ";
188.     cin >> Index;
189.     A.remove(Index);
190.     break;
191. }
192. case 8:
193. {
194.     cout << "Insert find value: ";
195.     cin >> Value;
196.     A.remove_by_value(Value);
197.     break;
198. }
199. }

200. } while (choose != 9);

201. system("pause");
202. return 0;
203. }
```

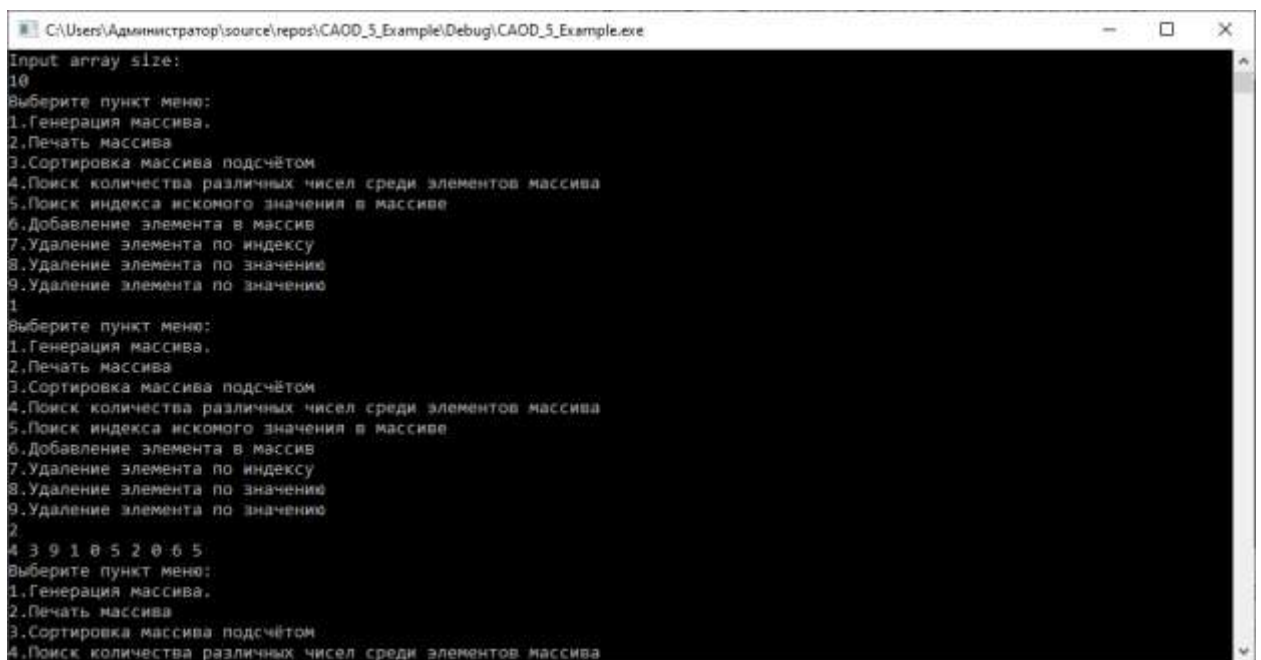
5. Пример выполнения программы

Ниже приведён пример выполнения программы



```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
Input array size:
10
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
```

Рис 1. Пример выполнения программы



```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
Input array size:
10
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
1
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
2
4 3 9 1 0 5 2 0 6 5
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
3
0 1 2 3 4 5 6 9
```

Рис 2. Пример выполнения программы


```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
2
4 3 9 1 0 5 2 0 6 5
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
3
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
2
0 0 1 2 3 4 5 5 6 9
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
```

Рис 3. Пример выполнения программы

```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
2
0 0 1 2 3 4 5 5 6 9
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
4
Уникальных элементов в массиве: 8
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
```

Рис 4. Пример выполнения программы

```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
4
Уникальных элементов в массиве: 8
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
5
Insert find value: 5
Индекс искомого элемента равен: 6
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
```

Рис 5. Пример выполнения программы

```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
5
Insert find value: 5
Индекс искомого элемента равен: 6
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
5
Insert find value: 2
Индекс искомого элемента равен: 3
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
```

Рис 6. Пример выполнения программы

```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
7
Insert index remove value: 3
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
2
0 0 1 3 4 5 5 6 9
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
```

Рис 7. Пример выполнения программы

```
C:\Users\Администратор\source\repos\CAOD_5_Example\Debug\CAOD_5_Example.exe
0 0 1 3 4 5 5 6 9
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
8
Insert find value: 9
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
5.Поиск индекса искомого значения в массиве
6.Добавление элемента в массив
7.Удаление элемента по индексу
8.Удаление элемента по значению
9.Удаление элемента по значению
2
0 0 1 3 4 5 5 6
Выберите пункт меню:
1.Генерация массива.
2.Печать массива
3.Сортировка массива подсчётом
4.Поиск количества различных чисел среди элементов массива
```

Рис 8. Пример выполнения программы