ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Анализ сложности алгоритмов |
| по курсу: СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

Целью работы является изучение методов и получение практических навыков анализа сложности алгоритмов.

**1.2 Задание на лабораторную работу**

Используя память, пропорциональную n, хранить массив целых чисел A, содержащий n элементов. Элементы массива A могут принимать случайные значения от -((n div 2) - 1) до (n div 2). То есть, если в массиве хранится 10 элементов, то эти элементы должны быть в диапазоне от -4 до 5. Разработать алгоритм, который осуществляет заполнение массива A случайными значениями, и по выбору пользователя выполняет одну из двух функций. Состав выполняемых функций и требования к теоретической временной сложности этих функций определяется вариантом задания. Варианты задания приведены в таблице 1.

**Вариант 17**

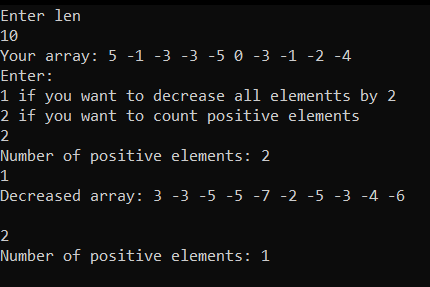


**Ход решения**

**Листинг**

1. #include <random>
2. #include <iostream>
3. #include <string>
4. using namespace std;
5. double fill() {
6. std::string input;
7. while (true) //проверка введенного числа
8. {
9. bool error = 0;
10. std::cin >> input;
11. for (int i = 0; i < input.size(); i++) {
12. if ((isdigit(input[i]) == 0 && input[i] != '.' && input[i] != '-')) {
13. error = 1;
14. break;
15. }
16. }
17. if (error == 1) {
18. printf("Enter value without letters\n");
19. }
20. else
21. {
22. break;
23. }
24. }
25. return stoi(input);
26. }
27. int\* gen\_array(int n)
28. {
29. random\_device rd;
30. mt19937 gen(rd());
31. int\* arr = new int[n];
32. for (int i = 0; i < n; i++)
33. {
34. int rand\_num = gen();
35. int pos = gen()%2;
36. if (pos == 1)
37. {
38. arr[i] = rand\_num % n/2+1;
39. }
40. else
41. {
42. arr[i] = -rand\_num % n/2-1;
43. }
44. }
45. return arr;
46. }
47. void show\_array(int arr[], int n)
48. {
49. for (int i = 0; i < n; i++)
50. {
51. cout << arr[i] << " ";
52. }
53. cout << endl;
54. }
55. int\* decrease\_by\_two(int arr[], int n)
56. {
57. for (int i = 0; i < n; i++)
58. {
59. arr[i] = arr[i] - 2;
60. }
61. return arr;
62. }
63. int count\_pos(int b)
64. {
65. if (b >= 0) {
66. return 1;
67. }
68. else {
69. return 0;
70. }
71. }
72. int main()
73. {
74. cout << "Enter len "<<endl;
75. int n = fill();
76. int\* arr = gen\_array(n);
77. cout << "Your array: ";
78. show\_array(arr, n);
79. cout << "Enter:\n1 if you want to decrease all elementts by 2\n2 if you want to count positive elements\n";
80. while (1) {
81. int input = fill();
82. ;
83. if (input == 1)
84. {
85. decrease\_by\_two(arr, n);
86. cout << "Decreased array: ";
87. show\_array(arr, n);
88. cout << endl;
89. }
90. else if (input == 2) {
91. int c = 0;
92. for (int i = 0; i < n; i++) {
93. if (count\_pos(arr[i]) == 1) {
    1. c++;
94. }
95. }
96. cout << "pos " << c << endl;
97. }
98. else {
99. break;
100. }
101. }
102. }

**Результат работы программы**



**Подсчет теоретических сложностей**

Алгоритм использует следующие данные (если не учитывать проверку введенных значений):

-4 переменных целого типа

-1 целочисленный массив размера n (ввод с клавиатуры)

**Пространственная сложность определяется следующей формулой:**

= n\*4 + 8

Где Cint – объем целочисленной переменной

**Теоретическая пространственная сложность составляет**

*V(n) = O(v) = O(*max*(O(n\*Cint)*, *O(2\*Cint))) = O(max(O(n), O(1))) = O(n)*

**Временная сложность** определяется на основе анализа текста программы.

**Теоретическая временная сложность функции**

\*Если пренебречь while 1 и считать, что за один запуск программы мы можем выбрать одно действие, иначе алгоритм будет выполняться, пока пользователь сам его не остановит, выбрав другое действие.

**Вывод**

На основе этих данных можно сказать, что был создан алгоритм, удовлетворяющий условиям варианта