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 алгоритмыобработки данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Иванов И.В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

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

**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.

**Вариант 7**

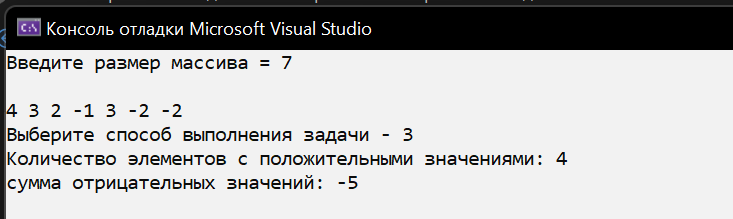
Подсчитать сумму всех элементов, имеющих отрицательные значения O(n)

Подсчитать количество элементов с положительными значениями O(1)

**Программный код:**

1. #include <iostream>
2. #include <ctime>
3. #include <cmath>
4. using namespace std;
5. int task1(int n, int arr[]) {
6. int sum = 0;
7. cout << "сумма отрицательных значений: ";
8. for (int i = 0; i < n; i++) {
9. if (arr[i] < 0) {
10. sum = sum + arr[i];
11. }
12. }
13. cout << sum << endl;
14. return sum;
15. }
16. void task2(int score) {
17. cout << "Количество элементов с положительными значениями: " << score;
18. }
19. int main() {
20. setlocale(LC\_ALL, "ru");
21. srand(time(0));
22. int n = 0;
23. int pick;
24. int score = 0;
25. cout << "Введите размер массива = ";
26. cin >> n;
27. int\* arr = new int[n];
28. for (int i = 0; i < n; i++) {
29. arr[i] = rand() % n - ((n / 2) - 1);
30. }
31. cout << endl;
32. for (int i = 0; i < n; i++) {
33. cout << arr[i] << ' ';
34. if (arr[i] >=0) {
35. score++;
36. }
37. }
38. cout << endl;
39. cout << "Выберите способ выполнения задачи - ";
40. cin >> pick;
41. switch (pick) {
42. case 1:
43. task1(n, arr);
44. break;
45. case 2:
46. task2(score);
47. break;
48. case 3:
49. task2(score);
50. cout << endl;
51. task1(n, arr);
52. break;
53. default:
54. cout << "Error!";
55. break;
56. }
57. delete[]arr;
58. return 0;
59. }

**Запуск программы:**



Теперь подсчитываем теоретические сложности алгоритма. Разработанный алгоритм использует следующие данные:

− одну константу;

− один массив размерностью n;

− пять переменных целого типа.

Значит, пространственная сложность алгоритма определяется следующим образом:

где – константа, характеризующая объем памяти, отводимый под переменную целого типа.

Теоретическая пространственная сложность алгоритма составляет:

Временную сложность алгоритма определяем на основе анализа текста программы, реализующей этот алгоритм. Согласно заданию на 6 лабораторную работу, необходимо реализовать две функции, теоретические временные сложности которых не превышают заданных. Поэтому необходимо рассчитать теоретическую временную сложность функции, реализующих эти алгоритмы:

+

где – константа, характеризующая время выполнения операций, помеченных i; и – временные сложности функций task1, task2 и всего алгоритмов целом, соответственно.

Теоретическая временная сложность функций составляет:

**Вывод:**

На основе этих расчетов можно сделать вывод, что был разработан алгоритм, характеристики которого соответствуют поставленному заданию.