

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

К.А. Кочин

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

АНАЛИЗ СЛОЖНОСТИ АЛГОРИТМОВ

по курсу: Структуры и алгоритмы обработки данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.

инициалы, фамилия

Санкт-Петербург 2022

Цель работы

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

Задание

Вариант №18.

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:

1. Все четные значения элементов уменьшить в два раза $O(n)$.
2. Подсчитать количество элементов с отрицательными значениями $O(1)$.

Листинг программы

```

1  #include <iostream>
2
3  using namespace std;
4
5  const int n = 10;
6
7  int r(int min, int max)
8  {
9      int num = min + rand() % (max - min + 1);
10     return num;
11 }
12
13 void f1(int a[n])
14 {
15     for (int i = 0; i < n; i++) {
16         if (a[i] % 2 == 0) {
17             a[i] = a[i] / 2;
18         }
19     }
20 }
21
22 int f2(int count)
23 {
24     return count;
25 }
26
27 int main()
28 {
29     setlocale(LC_ALL, "Rus");
30     srand(time(NULL));
31     int count, m, n;
32     count = 0;
33     cout << "Введите количество элементов массива ";
34     cin >> n;
35     int *a = new int [n];
36     cout << "Исходный массив: " << endl;
37     for (int i = 0; i < n; i++) {
38         a[i] = r(-(n / 2) - 1, (n / 2));
39         if (a[i] < 0) {
40             count++;
41         }
42         cout << a[i] << endl;
43     }

```

```

43     }
44     cout << "Если хотите все четные значения элементов уменьшить в дв
45     cin >> m;
46     cout << endl;
47     if ((m != 1) && (m != 2)) {
48         cout << "Error" << endl;
49     }
50     if (m == 1) {
51         f1(a);
52         for (int i = 0; i < n; i++) {
53             cout << a[i] << endl;
54         }
55     }
56     if (m == 2) {
57         cout << f2(count) << endl;
58     }
59     if (m == 0) {
60         cout << "Thanks" << endl;
61     }
62
63     return 0;
64 }

```

Расчет сложности алгоритма

Теперь подсчитываем теоретические сложности алгоритма.

Разработанный алгоритм использует следующие данные:

- одну константу;
- один массив размерностью n;
- четыре переменных целого типа.

Значит, пространственная сложность алгоритма определяется следующим образом:

$$V = C_{\text{const}} + n * C_{\text{int}} + 4 * C_{\text{int}}$$

где C_{const} – константа, характеризующая объем памяти, отводимый под константу; C_{int} – константа, характеризующая объем памяти, отводимый под переменную целого типа. Теоретическая пространственная сложность алгоритма составляет:

$$V(n) = O(v) = O(\max(O(C_{\text{const}}), O(n * C_{\text{int}}), O(4 * C_{\text{int}}))) = O(\max(O(1), O(n), O(1))) = O(n)$$

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

$$tf1 = n (K20 + K19 + K21)$$

$$tf2 = K15$$

$$tAlg = K25 + n(K30+K29+K32) + K34 + K39 + K38 + K39 + K40 + K41 + tf1 + K43 + n*(K44 + K45) + K42 + n*(K50+K50+K51)) + K52 + tf2$$

где K_i – константа, характеризующая время выполнения операций, помеченных i ; $tf1$ $tf2$ и $tAlg$ – временные сложности функций и всего алгоритма в целом, соответственно. Теоретическая временная сложность функций составляет:

$$Tf1(n) = O(tf1) = O(n)$$

$$Tf2(n) = O(tf2) = O(1)$$

Вывод

В ходе выполнения работы были получены навыки расчета временной сложности алгоритмов. На основе этих расчетов можно сделать вывод, что был разработан алгоритм, характеристики которого соответствуют поставленному заданию.