

11. ПРАКТИЧЕСКИЕ ЗАДАНИЯ

11.1. Практическая работа 1

Тема: «Оценка сложности и определение эффективности алгоритма».

Задание 1. Определить эффективный алгоритм из двух предложенных, используя оценку теоретической сложности каждого из алгоритмов и ёмкостную сложность.

Вычислительная задача: дан массив из n элементов целого типа, удалить из массива все значения равные заданному.

Удаление состоит в уменьшении размера массива и сохранения порядка следования всех элементов, как до удаляемого, так и следующих после удаляемого.

Например, необходимо удалить из массива все значения равны 2.

Исходный массив: $n=10$; 1 2 3 2 2 2 5 2 2 2. Результат: $n=3$; 1 3 5.

Таблица 11. Алгоритмы решения задачи

x-массив, n – количество элементов в массиве, key – удаляемое значение	
Алгоритм 1. delFirstMetod(x,n,key){ $i \leftarrow 1$ while ($i \leq n$) do if $x[i]=key$ then //удаление for $j \leftarrow i$ to $n-1$ do $x[j] \leftarrow x[j+1]$ od $n \leftarrow n-1$ else $i \leftarrow i+1$ endif od }	Алгоритм 2 delOtherMetod(x,n,key){ $j \leftarrow 1$ for $i \leftarrow 1$ to n do $x[j]=x[i]$; if $x[i] \neq key$ then $j++$ endif od $n \leftarrow j$ }

Требования к выполнению задания:

1. Для алгоритма привести этапы разработки:

1.1. Постановка задачи

1.2. Модель решения поставленной задачи.

а) Описать, как выполняется алгоритм (словами и блок-схемой).

б) Определить для внешнего цикла инвариант цикла – доказать корректность цикла.

с) Определить вычислительную сложность алгоритма используя теоретический подход.

- 1.3. Реализовать алгоритм в виде функции и отладить на массиве при $n=10$, $n=100$. Включить в функцию операторы, подсчитывающие число выполненных сравнений и перемещений элементов при удалении.
 - 1.4. Реализовать функции: заполнение массива датчиком случайных чисел, вывод массива на экран монитора.
 - 1.5. Представить результаты тестирования, указав количество операций согласно теоретическим расчетам и полученным при выполнении алгоритма.
 - 1.6. Протестировать алгоритм в случаях: все элементы должны быть удалены, ни один элемент не удаляется. Сравнить результаты теоретической сложности этих случаев.
2. Включите в отчет результаты по заданию 1, в т.ч. ответы на пункты 1.1 до 1.6. для каждого алгоритма в отдельности.

Задание 2. Индивидуальная задача

1. Выполнить разработку программы в соответствии с задачей варианта, включив в разработку следующие этапы:
 - 1.1. Постановка задачи
 - 1.2. Модель решения
 - 1.3. Разработка **эффективного** алгоритма
 - а) разработать алгоритм
 - б) определить инвариант
 - в) доказать корректность циклов в алгоритме
 - г) определить вычислительную сложность алгоритма на основе теоретического подхода
 - 1.4. Реализовать алгоритм варианта в виде одной функции (без декомпозиции на другие функции).
 - 1.5. Провести тестирование алгоритма на массиве из 10 чисел. Для этого разработать таблицу тестов и включить набор тестов в соответствии с ограничениями постановки задачи. Выполнить тестовые прогоны и убедиться, что все требования выполняются.
 - 1.6. Выполнить практическую оценку сложности алгоритма для больших n . Показать результаты прогонов для заданного n в лучшем и худшем случаях.
2. Включите в отчет результаты по заданию 2, отобразив описание выполнения всех этапов с 1.1 по 1.6. и код всей программы со скринами результатов тестирования.

Таблица 12. Варианты индивидуальных заданий

№ варианта	Задача
0	Найти количество натуральных чисел, не превосходящих заданного n и делящихся на каждую из своих цифр.
1	Умножение квадратных матриц.
2	Умножение матрицы на вектор.
3	Сложение двух матриц
4	Получение матрицы обратной данной матрице
5	Обход матрицы по спирали (по часовой стрелке: первая строка, последний столбец, нижняя строка, первый столбец)
6	Найти максимальный элемент в части матрицы, расположенной над главной диагональю.
7	Найти минимальное четное число в части матрицы – между главной и побочной диагоналями (диагонали образуют вертикальные песочные часы).
8	Найти восходящую диагональ матрицы с максимальной суммой элементов.
9	Определить, симметрична ли матрица относительно главной диагонали.
10	Выполнить транспонирование матрицы
11	Дан одномерный массив из n элементов целого типа. Определить, сколько раз в массив входит максимальное значение.
12	Реализовать алгоритм «схема Горнера» вычисления значения линейного многочлена n -ой степени.
13	Дана прямоугольная матрица размером $n \times m$. Определить максимальное из чисел, встретившихся в матрице более одного раза.
14	Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы размером $n \times (n+1)$. С помощью допустимых преобразований привести систему к треугольному виду (коэффициенты должны быть только над главной диагональю). Примечание. Система состоит из n уравнений с n неизвестными. Матрица имеет размер $n \times (n+1)$. Т.е. i -ая строка матрицы хранит коэффициенты i -ого уравнения и свободный член.
15	Дана целочисленная прямоугольная матрица размером $n \times m$. Характеристикой строки матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.
16	Дана целочисленная квадратная матрица размером $n \times n$. Найти минимум среди сумм модулей элементов диагоналей параллельных побочной диагонали.