

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет "Высшая школа экономики"**

Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ
Департамент компьютерной инженерии

Курс: Алгоритмизация и программирование

Раздел	Макс. оценка	Итоговая оценка
Постановка	0.5	
Метод	1	
Спецификация	0.5	
Алгоритм	1.5	
Работа программы	1	
Листинг	0.5	
Тесты	1	
Вопросы	2	
Доп. задание	2	

Отчет по лабораторной работе № 2

Студент: Чапайкин Арсений Георгиевич

Группа: БИВ242

Вариант: № 171 (1, 1)

Руководитель: Елисеенко А.М.

Оценка:

Дата сдачи:

Содержание

Задание	2
Постановка задачи	3
Метод решения	4
Внешняя спецификация	5
Описание алгоритма на псевдокоде	6
Листинг программы	10
Распечатка тестов к программе и результатов	14

Задание

1. Для данных n, m вычислить $a = \exp\{\ln(m!) - \ln(n!) - \ln((n - m)!)\}$.
2. Для данной матрицы $C[1:n; 1:m]$ вычислить $V = \min_{i=1\dots n} \{ \max_{j=1\dots m-1} |c_{i,j} - c_{i,j+1}| \}$.
3. Для данного массива $A[1:n]$:
 - (a) Подсчитать для каждого элемента количество раз, которое он встречается в A .
 - (b) Удалить из A все неповторяющиеся элементы.

Постановка задачи

Дано:

1. $n, m \in \mathbb{N}$.
2. $n, m \in \mathbb{N}$. $C_{i,j} \in \mathbb{Z}$.
3. $n \in \mathbb{N}$. $A_i \in \mathbb{Z}$.

Результат:

1. $a \in \mathbb{N}$.
2. $V \in \mathbb{R}$.
3. $A[1:k]$ — целочисленный массив.

При:

2. $0 < n, m \leq \text{MaxSize}$.
3. $0 < n \leq \text{MaxSize}$.

Связь:

1. См. формулу в условии.
2. См. формулу в условии.
3. (a) $\forall i = \overline{1 \dots n} \text{ cnt}[i] = |\{j \in [1, n] \mid A[j] = A[i]\}|$.
(b) Для $i = \overline{1 \dots n} : \text{cnt}[i] \neq 1$ требуется найти $j = \min\{v \in [1, i] \mid \text{cnt}[v] = 1 \vee v = i\}$ и присвоить $A[j] := A[i]$.

Метод решения

1. $numerator := 1$

$$\begin{cases} \text{Для } i = \overline{m - n + 1 \dots m} \\ numerator = numerator \cdot i \end{cases}$$

$y := 1$

$$\begin{cases} \text{Для } i = \overline{2 \dots n} \\ y = y \cdot i; \end{cases}$$

$$a = \frac{numerator}{y}$$

2. $result := IntMax$

$$\begin{cases} \text{Для } i = \overline{1 \dots n} \\ maximum := IntMin \\ \begin{cases} \text{Для } j = \overline{1 \dots m - 1} \\ maximum = \max(maximum, |C[i][j] - C[i][j + 1]|) \end{cases} \\ result = \min(result, maximum) \end{cases}$$

$$3. \begin{cases} \text{Для } i = \overline{1 \dots n} \\ \begin{cases} \text{Для } j = \overline{1 \dots n} \\ (cnt[i] = cnt[i] + 1)), \text{ если } (A[i] = A[j]) \end{cases} \end{cases}$$

$pos := 0$

$$\begin{cases} \text{Для } i = \overline{1 \dots n} \\ (A[k] = A[i], pos = pos + 1), \text{ если } cnt[i] \neq 1 \end{cases}$$

Внешняя спецификация

Задание 1

Task 1

Input two positive integers: n m

$\left\{ \begin{array}{l} \langle n \rangle \langle m \rangle \\ \text{Input format is incorrect. Try again.} \end{array} \right\}^*$ До $n, m \in \mathbb{N} \wedge n \leq m$

Задание 2

Task 2

Input matrix dimensions in the following format:

'number of rows' 'number of columns'

Number of columns must be greater than 1

$\left\{ \begin{array}{l} \langle n \rangle \langle m \rangle \\ \text{Input format is incorrect. Try again.} \end{array} \right\}^*$ До $n, m \in \mathbb{N} \wedge n \leq MaxSize \wedge 1 < m \leq MaxSize$

Result: «a»

Input matrix in the following format:

C[1][1] C[1][2] ... C[n][m]

$\left\{ \begin{array}{l} \langle C[1][1] \rangle \langle C[1][2] \rangle \dots \langle C[n][m] \rangle \\ \text{Input format is incorrect. Try again.} \end{array} \right\}^*$ До $\forall i = \overline{1 \dots n} \forall j = \overline{1 \dots m} C[i][j] \in \mathbb{Z}$

Result: «V»

Задание 3

Task 3

Input array size

$\left\{ \begin{array}{l} \langle n \rangle \\ \text{Input format is incorrect. Try again.} \end{array} \right\}^*$ До $n \in \mathbb{N} \wedge n \leq MaxSize$

Input array in the following format:

A[1] A[2] ... A[n]

$\left\{ \begin{array}{l} \langle A[1] \rangle \langle A[2] \rangle \dots \langle A[k] \rangle \\ \text{Input format is incorrect. Try again.} \end{array} \right\}^*$ До $\forall i = \overline{1 \dots n} A[i] \in \mathbb{Z}$

при $n = 0$

$\left\{ \begin{array}{l} \text{Resulting array:} \\ A[1] A[2] \dots A[n] \end{array} \right\}$

иначе

$\left\{ \begin{array}{l} \text{Resulting array is empty} \end{array} \right\}$

Описание алгоритма на псевдокоде

Задание 1

функция factorial(x)

 result := 1

цикл i := 2 до x:

 result = result · i

кц

кц

Нач

 вывод("Задача 1")

 вывод("Введите два положительных целых числа: n m")

 n, m: целое положительное число

цикл-пока (ввод(n) не целое положительное) или (ввод(m) не целое положительное) или (m == 0) или (n == 0) или (n > m):

 вывод("Неверный формат ввода. Попробуйте снова.")

кц

 numerator := 1

цикл i := m - n + 1 до m:

 numerator = numerator · i

кц

 вывод(Результат: $\frac{numerator}{factorial(n)}$)

Конец

Задание 2

Константа MaxSize = 100

функция validSize(size: целое число):

вернуть size > 0 и size <= MaxSize

кц

процедура inputArray(size: целое число, array: массив целых чисел):

 correctInput: булево

цикл:

 correctInput = true

цикл i := 0 до size - 1:

 ввод(array[i])

если array[i] не является целым числом то

 correctInput = false

всё

кц

если !correctInput то

 вывод("Неверный формат ввода. Попробуйте снова.")

всё

```

    до correctInput
    кц
кц

Нач
    вывод("Задача 2")
    вывод("Введите размеры матрицы в следующем формате: 'количество строк' 'количество столбцов'")
    вывод("Количество столбцов должно быть больше 1")

    rows, cols: целое положительное число
    цикл-пока (ввод(rows) не целое) или (ввод(cols) не целое) или !validSize(rows) или !validSize(cols):
        вывод("Неверный формат ввода. Попробуйте снова.")
    кц

    вывод("Введите матрицу в следующем формате:")
    вывод("matrix[0][0] matrix[0][1] ... matrix[rows][cols]")

    matrix := [1 : rows · cols]
    inputArray(rows · cols, matrix)

    result := ∞
    цикл i := 0 до rows - 1:
        maximum := -∞
        цикл j := 0 до cols - 2:
            maximum = max(maximum, |matrix[i · cols + j] - matrix[i · cols + j + 1]|)
        кц
        result = min(result, maximum)
    кц
    вывод(Результат: result)
Конец

```

Задание 3

Константа MaxSize = 100

```

функция validSize(size: целое число):
    вернуть size > 0 и size <= MaxSize
кц

```

```

процедура inputArray(size: целое число, array: массив целых чисел):
    correctInput: булево
    цикл:
        correctInput = true
        цикл i := 0 до size - 1:
            ввод(array[i])
            если array[i] не является целым числом то
                correctInput = false
        всё
    кц

```



```

        если !correctInput то
            вывод("Неверный формат ввода. Попробуйте снова.")
        всё
    до correctInput
    кц
кц

```

```

функция count(length, array)
    cnt := [1 : length]
    цикл i := 0 до length - 1:
        цикл j := 0 до length - 1:
            если array[i] == array[j] то
                cnt[i] = cnt[i] + 1
            кц
        кц
    кц
    вернуть cnt
кц

```

```

функция deleteUnique(length, array)
    cnt := count(length, array)
    lastFreePos := 0
    цикл i := 0 до length - 1:
        если cnt[i] != 1 то
            array[lastFreePos] = array[i]
            lastFreePos = lastFreePos + 1
        кц
    кц
кц

```

```

Нач
    вывод("Задача 3")
    вывод("Введите размер массива")

    length: целое положительное число
    цикл-пока (ввод(length) не целое) или !validSize(length):
        вывод("Неверный формат ввода. Попробуйте снова.")
    кц

    вывод("Введите элементы массива (целые числа) в следующем формате:")
    вывод("array[0] ... array[sizeofarray]")

    array := [1 : length]
    inputArray(length, array)
    deleteUnique(length, array)

    если length != 0 то
        вывод("Результирующий массив:")
        цикл i := 0 до length - 1:
            вывод(array[i])
    кц

```

иначе

вывод("Результирующий массив пуст")

кц

Конец

Листинг программы

Задание 1

```
1 #include <stdio.h>
2 #include <assert.h>
3
4 unsigned long long factorial(unsigned x) {
5     unsigned long long result = 1;
6     for (int i = 2; i <= x; i++) {
7         result *= i;
8     }
9     assert((result != 0) && "n! is too large");
10    return result;
11 }
12
13 void input(unsigned* n, unsigned* m) {
14     while(scanf("%u %u", n, m) != 2 || *n == 0 || *m == 0 || *n > *m) {
15         puts("Wrong input format. Try again");
16     }
17 }
18
19 int main() {
20     puts("Task 1");
21     puts("Input two positive integers: n m");
22
23     unsigned n, m;
24     input(&n, &m);
25
26     unsigned long long numerator = 1;
27     for (int i = m - n + 1; i <= m; i++) {
28         numerator *= i;
29     }
30     assert((numerator != 0) && "Numerator is too large");
31
32     printf("Result: %llu", numerator / factorial(n));
33 }
```

Задание 2

```
1 #include <stdio.h>
2 #include <limits.h>
3 #include <stdbool.h>
4 #include <math.h>
5 #include <stdlib.h>
6
7 const size_t MaxSize = 100;
8
9 int max(int a, int b) { return (a > b ? a : b); }
10 int min(int a, int b) { return (a < b ? a : b); }
11
12 bool valid_size(size_t size) {
13     return 0 < size && size <= MaxSize;
14 }
15
16 void input_array(size_t size, int* array) {
17     bool correct_input;
18     do {
```

```

19     correct_input = true;
20     for (size_t i = 0; i < size; i++) {
21         if (scanf("%d", &array[i]) != 1) {
22             correct_input = false;
23         }
24     }
25     if (!correct_input) {
26         ("Wrong input format. Try again.");
27     }
28 } while (!correct_input);
29 }
30
31 void input_dimensions(size_t *rows, size_t *cols) {
32     while (scanf("%zu%zu", rows, cols) != 2 || !valid_size(*rows) || !valid_size(*
33         cols) || *cols < 2) {
34         puts("Wrong input format. Try again.");
35     }
36 }
37
38 int main() {
39     puts("Task 2");
40     puts("Input matrix dimensions in the following format:\
41 'number of rows' 'number of columns'");
42     puts("Number of columns must be greater than 1");
43
44     size_t rows, cols;
45     input_dimensions(&rows, &cols);
46
47     puts("Input matrix in the following format:");
48     puts("matrix[0][0] matrix[0][1] ... matrix[1][1] ... matrix[rows][cols]");
49
50     int* matrix = (int*)malloc(rows * cols * sizeof(int));
51     input_array(rows * cols, matrix);
52
53     int result = INT_MAX;
54     for (size_t i = 0; i < rows; i++) {
55         int maximum = INT_MIN;
56         for (size_t j = 0; j + 1 < cols; j++) {
57             maximum = max(maximum, abs(matrix[i * rows + j] - matrix[i * rows + j +
58                 1]));
59         }
60         result = min(result, maximum);
61     }
62
63     printf("Result: %d\n", result);
64     return 0;
65 }

```

Задание 3

```

1 #include <stdio.h>
2 #include <stdbool.h>
3 #include <stdlib.h>
4
5 const size_t MaxSize = 100;
6
7 int* count(size_t length, int *array) {
8     int* cnt = (int*)malloc(length * sizeof(int));

```

```

9     for (size_t i = 0; i < length; i++) {
10         for (size_t j = 0; j < length; j++) {
11             if (array[i] == array[j]) {
12                 cnt[i]++;
13             }
14         }
15     }
16     return cnt;
17 }
18
19 size_t delete_unique(size_t length, int *array) {
20     int *cnt = count(length, array);
21     size_t lastFreePos = 0;
22
23     for (size_t i = 0; i < length; i++) {
24         if (cnt[i] != 1) {
25             array[lastFreePos++] = array[i];
26         }
27     }
28     array = realloc(array, lastFreePos * sizeof(int));
29     return lastFreePos;
30 }
31
32 bool valid_size(size_t size) {
33     return 0 < size && size <= MaxSize;
34 }
35
36 void input_array(size_t size, int* array) {
37     bool correct_input;
38     do {
39         correct_input = true;
40         for (size_t i = 0; i < size; i++) {
41             if (scanf("%d", &array[i]) != 1) {
42                 correct_input = false;
43             }
44         }
45         if (!correct_input) {
46             ("Wrong input format. Try again.");
47         }
48     } while (!correct_input);
49 }
50
51 void output(size_t new_length, int *array) {
52     if (new_length != 0) {
53         puts("Resulting array:");
54         for (size_t i = 0; i < new_length; i++) {
55             printf("%d ", array[i]);
56         }
57     } else {
58         puts("Resulting array is empty");
59     }
60 }
61
62 int main() {
63     puts("Task 3");
64     puts("Input array size");
65
66     size_t length;
67     while (scanf("%zu", &length) != 1 || !valid_size(length)) {
68         puts("Wrong input format. Try again");

```

```
69     }
70
71     puts("Input array elements (integers) in the following format:");
72     puts("array[0] ... array[size_of_array]");
73
74     int* array = (int*)malloc(length * sizeof(int));
75     input_array(length, array);
76
77     size_t new_length = delete_unique(length, array);
78     output(new_length, array);
79     return 0;
80 }
```

Распечатка тестов к программе и результатов

Задание 1

Номер	Исходные данные	Результат
1	n = 1, m = 2	Result: 2
2	n = 3, m = 5	Result: 10
3	n = 7, m = 11	Result: 330
4	n = 10, m = 15	Result: 3003
5	n = 100, m = 200	Error: n! is too large

Задание 3

Номер	Исходные данные	Результат
1	n = 5 A = 1 2 3 4 5	Resulting array is empty
2	n = 6 A = 1 1 2 2 3 3	Resulting array: 1 1 2 2 3 3
3	n = 7 A = 1 -2 5 7 1 0 -2	Resulting array: 1 -2 1 -2

Задание 2

Номер	Исходные данные	Результат
1	n = 3, m = 3 C = 1 2 3 4 5 6 7 8 9	Result: 1
2	n = 3, m = 3 C = 0 0 0 0 0 0 0 0 0	Result: 0
3	n = 3, m = 3 C = -1 -2 -3 -4 -5 -6 -7 -8 -9	Result: 1
4	n = 3, m = 3 C = 111 237 14 1026 51 999 228 69 666	Result: 223
5	n = 3, m = 3 C = 3 3 3 6 6 6 9 9 9	Result: 0