МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК  
  
Кафедра прикладной математики, механики, управления и программного обеспечения**

**ОТЧЁТ ПО ИДЗ №2 ПО ДИСЦИПЛИНЕ  
СТРУКТУРЫ И АЛГОРИТМЫ КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ**

Направление 02.03.03 «Математическое обеспечение и администрирование  
информационных систем»

Выполнил студент гр. Б8204  
Скопина Александра Артуровна  
Проверил:  
Доцент кафедры С.Н. Остроухова

Владивосток  
2018 г.

**Формальная постановка задачи:**

**X = {**M – матрица n×n: n > 1, Mi, j ∈ ; k ∈ **}**

**Y = {**S = {S1, S2, S3}

S1=” Вектор: A. Последовательность не отсортирована.”

S2=” Вектор: A. Ключ не найден.”

S3 =” Вектор: A. (Бинарный): [i1,j1]. (Однородный бинарный): [i2,j2]” ,   
где [i1,j1], [i2,j2] – номера элемента матрицы M, равного k.

[i1,j1] – найден с помощью бинарного поиска.

[i2,j2] – найден с помощью однородного бинарного поиска.**}**

**F = {**a – диагональ, параллельная главной, p – длина вектора a.

Если , то S1, иначе:

* Если , то S2
* Если , то S3 **}**

**Спецификация типов:** type vector = array [1..max] of word; // вектор

**Спецификация функций:**

1. function BiSearch (const a: vector; const n: byte; const key: word):integer;

Ищет в векторе a длины n элемент key, используя бинарный поиск. Возвращает:

* -1, если поступившая на вход последовательность не отсортирована по возрастанию
* 0, если элемент key не найден
* Если в a найден key, то возвращает его порядковый номер

1. function UniformBiSearch (const a: vector; const n: byte; const key: word):integer;

Ищет в векторе a длины n элемент key, используя однородный бинарный поиск. Возвращает:

* -1, если поступившая на вход последовательность не отсортирована по возрастанию
* 0, если элемент key не найден
* Если в a найден key, то возвращает его порядковый номер

**Тесты:**

1. function BiSearch (const a: vector; const n: byte; const key: word):integer;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Описание тестовой ситуации** | **Входные данные** | | | **Выходные данные** |
|  | a | n | key |  |
| Неотсортированная последовательность | 1 2 6 4 5 | 5 | 6 | -1 |
| Ключ отсутствует в последовательности | 1 2 2 3 3 | 5 | 6 | 0 |
| 5 5 5 6 6 6 | 6 | 2 | 0 |
| 1 2 3 5 6 | 5 | 4 | 0 |
| Последовательность отсортирована и ключ найден | 1 2 3 4 5 6 7 8 9 10 11 | 11 | 6 | 6 |
| 0 6 6 8 9 9 9 9 9 9 | 10 | 6 | 2 |
| 6 6 6 6 6 6 6 6 6 6 | 10 | 6 | 5 |
| 0 0 0 0 0 0 6 6 20 | 9 | 6 | 7 |
| 6 6 6 8 10 20 30 30 40 | 9 | 6 | 2 |
| 0 0 0 0 2 6 6 6 | 8 | 6 | 6 |

1. function UniformBiSearch (const a: vector; const n: byte; const key: word):integer;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Описание тестовой ситуации** | **Входные данные** | | | **Выходные данные** |
|  | a | n | key |  |
| Неотсортированная последовательность | 1 2 6 4 5 | 5 | 6 | -1 |
| Ключ отсутствует в последовательности | 1 2 2 3 3 | 5 | 6 | 0 |
| 5 5 5 6 6 6 | 6 | 2 | 0 |
| 1 2 3 5 6 | 5 | 4 | 0 |
| Последовательность отсортирована и ключ найден | 1 2 3 4 5 6 7 8 9 10 11 | 11 | 6 | 6 |
| 0 6 6 8 9 9 9 9 9 9 | 10 | 6 | 2 |
| 6 6 6 6 6 6 6 6 6 6 | 10 | 6 | 5 |
| 0 0 0 0 0 0 6 6 20 | 9 | 6 | 7 |
| 6 6 6 8 10 20 30 30 40 | 9 | 6 | 3 |
| 0 0 0 0 2 6 6 6 | 8 | 6 | 6 |