Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по лабораторной работе №6

Тема работы: Сортировка массивов

Выполнил

студент: гр. 551003 Дементей В.С.

Проверил: Фадеева Е.П.

Минск 2015

Содержание

[1 Постановка задачи 3](#_Toc438484074)

[2 Структура данных 4](#_Toc438484075)

[3 Текстовый алгоритм 5](#_Toc438484076)

[4 Схема алгоритма решения задачи по ГОСТ 19.701-90 7](#_Toc438484077)

[5 Схема алгоритма (представление по методу Насси-Шнейдермана) 10](#_Toc438484078)

[Приложение А 12](#_Toc438484079)

[Приложение Б 15](#_Toc438484080)

# Постановка задачи

Дана матрица Х[9,7]. Упорядочить элементы строк матрицы по неубы- ванию (одно число в упорядоченном (отсортированном) списке встанет раньше другого, если первое число меньше или равно второму.), а сами строки по убыванию минимальных элементов строк. (Вывести исходный массив, массив с упорядоченными элементами в строках, столбец минимальных элементов, массив упорядоченный по убыванию минимальных элементов строк)

# Структура данных

Таблица - Данные

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Примечание |
| col | integer | Количество столбцов | Константа |
| str | integer | Количество строк | Константа |
| Arr\_opening | Array [0..str,0..col] of integer | Массив чисел |  |
| i | integer | Счетчик цикла |  |
| j | integer | Счетчик цикла |  |
| k | integer | Счетчик цикла |  |
| temp | integer | Временное значение |  |

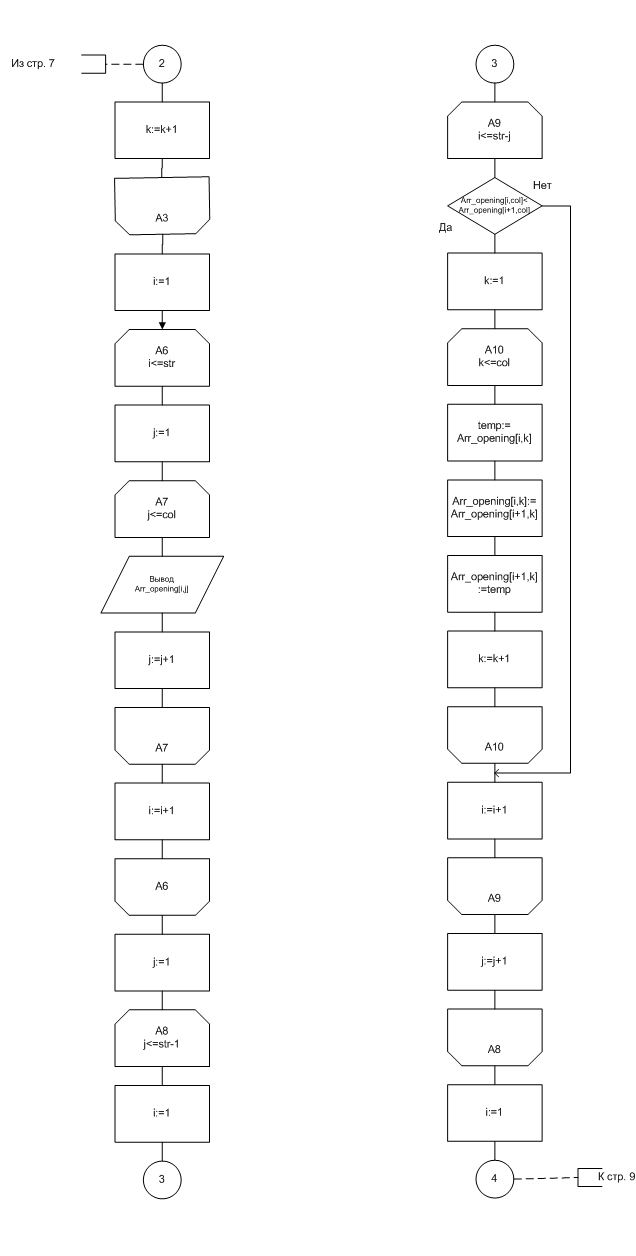
# Текстовый алгоритм

Таблица 2 - Текстовый алгоритм

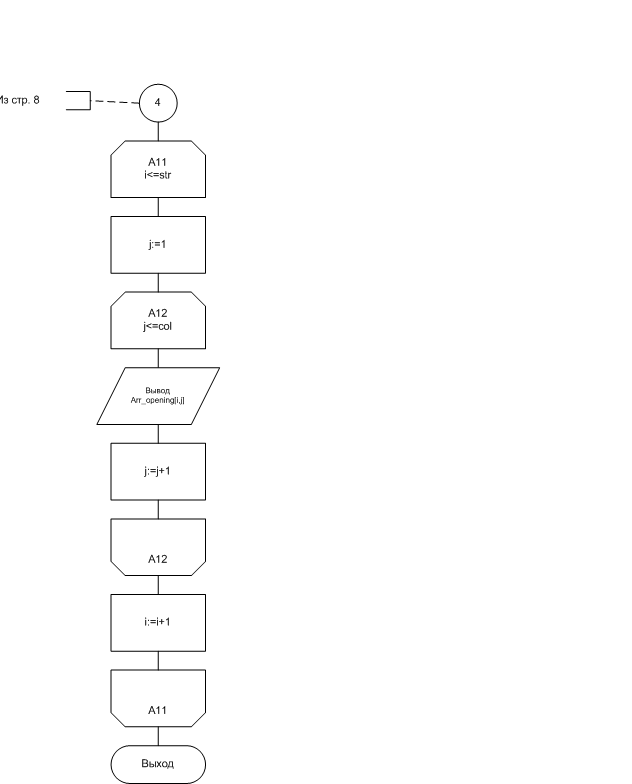
|  |  |
| --- | --- |
| Номер  шага | Назначение шага |
|  | str:=9 |
|  | col=8; |
|  | i:=1 |
|  | Начало цикла А1. Проверка условия (i<=str). Если оно истинно, то идти к шагу 5, иначе – к шагу 13. |
|  | j:=1 |
|  | Начало цикла А2. Проверка условия (j<=(col-1)). Если оно истинно, то идти к шагу 7, иначе – к шагу 11 |
|  | Arr\_opening[i,j]:=random(50) |
|  | Вывод значения Arr\_opening[i,j] |
|  | j:=j+1; |
|  | Конец цикла A2. Идти к шагу 6. |
|  | i:=i+1 |
|  | Конец цикла A1. Идти к шагу 4. |
|  | k:=1 |
|  | Начало цикла А3. Проверка условия (k<=str). Если оно истинно, то идти к шагу 15, иначе – к шагу 29. |
|  | i:=2 |
|  | Начало цикла А4. Проверка условия (i<=col-1). Если оно истинно, то идти к шагу 17, иначе – к шагу 26. |
|  | Arr\_opening[k,0]:=Arr\_opening[k,i] |
|  | j:=i-1 |
|  | Начало цикла А5. Проверка условия (Arr\_opening[k,j]>Arr\_opening[k,0]). Если оно истинно, то идти к шагу 20, иначе – к шагу 23. |
|  | Arr\_opening[k,j+1]:=Arr\_opening[k,j] |
|  | j:=j-1 |
|  | Конец цикла A5. Идти к шагу 19. |
|  | Arr\_opening[k,j+1]:=Arr\_opening[k,0] |
|  | i:=i+1 |
|  | Конец цикла A4. Идти к шагу 16. |
|  | Arr\_opening[k,col]:=Arr\_opening[k,1] |
|  | k:=k+1 |
|  | Конец цикла A3. Идти к шагу 14. |
|  | i:=1 |
|  | Начало цикла А6. Проверка условия (i<=str). Если оно истинно, то идти к шагу 31, иначе – к шагу 38. |
|  | j:=1 |
|  | Начало цикла А7. Проверка условия (j<=col). Если оно истинно, то идти к шагу 33, иначе – к шагу 36. |
|  | Вывод значения Arr\_opening[i,j] |
|  | j:=j+1 |
|  | Конец цикла A7. Идти к шагу 32. |
|  | i:=i+1 |
|  | Конец цикла A6. Идти к шагу 30. |
|  | j:=1 |
|  | Начало цикла А8. Проверка условия (j<=str-1). Если оно истинно, то идти к шагу 40, иначе – к шагу 54. |
|  | i:=1 |
|  | Начало цикла А9. Проверка условия (i<=str-j). Если оно истинно, то идти к шагу 42, иначе – к шагу 52. |
|  | Проверка выполнения условия (Arr\_opening[i,col]<Arr\_opening[i+1,col]). Если оно истинно, то идти к шагу 43, иначе – к шагу 50. |
|  | k:=1 |
|  | Начало цикла А10. Проверка условия (k<=col). Если оно истинно, то идти к шагу 45, иначе – к шагу 50. |
|  | temp:=Arr\_opening[i,k] |
|  | Arr\_opening[i,k]:=Arr\_opening[i+1,k] |
|  | Arr\_opening[i+1,k]:=temp |
|  | k:=k+1 |
|  | Конец цикла A10. Идти к шагу 44. |
|  | i:=i+1 |
|  | Конец цикла A9. Идти к шагу 41. |
|  | j:=j+1 |
|  | Конец цикла A8. Идти к шагу 39. |
|  | i:=1 |
|  | Начало цикла А11. Проверка условия (i<=str). Если оно истинно, то идти к шагу 56, иначе – к шагу 63. |
|  | j:=1 |
|  | Начало цикла А12. Проверка условия (j<=col). Если оно истинно, то идти к шагу 58, иначе – к шагу 61. |
|  | Вывод значения Arr\_opening[i,j] |
|  | j:=j+1 |
|  | Конец цикла A12. Идти к шагу 57. |
|  | i:=i+1 |
|  | Конец цикла A11. Идти к шагу 55. |
|  | Останов. |

# C:\Users\Mike\Desktop\Labs\Labs\6 сортировки\Алгоритм ГОСТ часть 1.pngСхема алгоритма решения задачи по ГОСТ 19.701-90

Рисунок 1 – Схема алгоритма (часть 1)

****

**Рисунок 2 – Схема алгоритма (часть 2)**

Рисунок 3 – Схема алгоритма (часть 3)

# Схема алгоритма (представление по методу насси-шнейдермана)

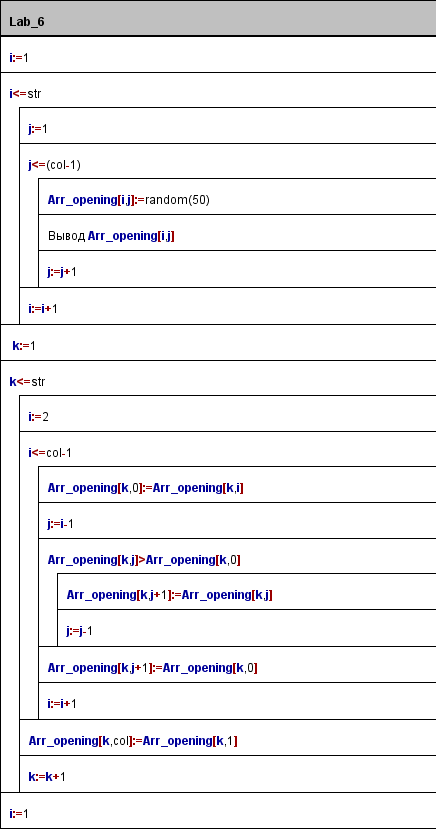


Рисунок 4 – Схема алгоритма по методу Насси-Шнейдермана (часть 1)

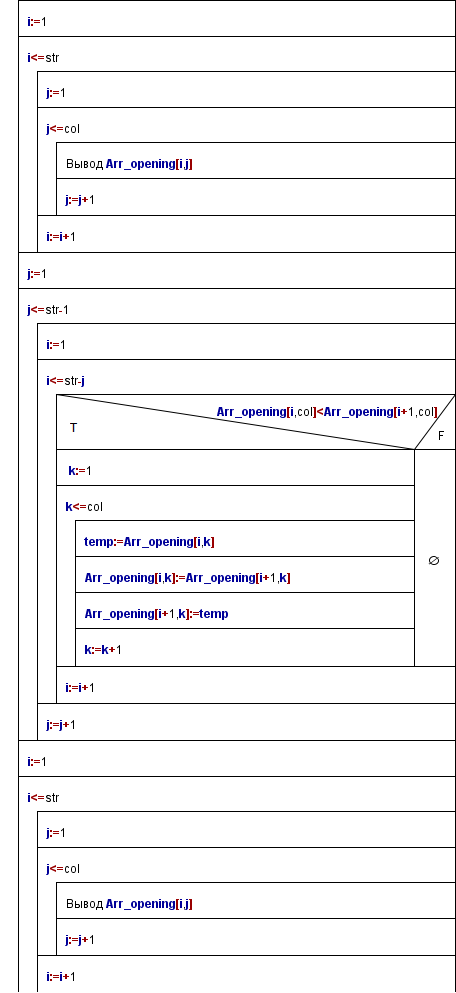


Рисунок 5 – Схема алгоритма по методу Насси-Шнейдермана (часть 2)

Приложение А

(обязательное)

Исходный код программы

**Program** Lab\_6;

{$APPTYPE CONSOLE}

**Uses**

SysUtils;

**Const**

str=9;

col=7+1;

**Var**

Arr\_opening:**array**[0..str,0..col] **of** integer;

i,j,k,temp:integer;

**Begin**

writeln('Initial data');

writeln('==================================');

{Создание массива заполненного случайными значениями [0;49]}

randomize;

i:=1;

**while**(i<=str) **do**

**begin**

j:=1;

**while**(j<=(col-1)) **do**

**begin**

Arr\_opening[i,j]:=random(50);

write(Arr\_opening[i,j]:3);

write(' ');

j:=j+1;

**end**;

writeln;

i:=i+1;

**end**;

writeln('==================================');

writeln;

writeln('First sorting');

writeln('==================================');

{Сортировка строк массива

методом простой вставки с барьерным элементом}

k:=1;

**while**(k<=str) **do**

**begin**

i:=2;

**while**(i<=col-1) **do**

**begin**

Arr\_opening[k,0]:=Arr\_opening[k,i];

j:=i-1;

{Сравнение чисел

Если <=, то младший элемент становится в нужную позицию}

**while** (Arr\_opening[k,j]>Arr\_opening[k,0]) **do**

**begin**

Arr\_opening[k,j+1]:=Arr\_opening[k,j];

j:=j-1;

**end**;

Arr\_opening[k,j+1]:=Arr\_opening[k,0];

i:=i+1;

**end**;

{Запись минимального элемента}

Arr\_opening[k,col]:=Arr\_opening[k,1];

k:=k+1;

**end**;

{Вывод значений массива после сортировки строк}

i:=1;

**while**(i<=str) **do**

**begin**

j:=1;

**while**(j<=col) **do**

**begin**

write(Arr\_opening[i,j]:3);

write(' ');

**if** j=col-1 **then**

write(' | ');

j:=j+1;

**end**;

i:=i+1;

writeln;

**end**;

writeln('==================================');

writeln;

writeln('The second sorting');

writeln('==================================');

writeln;

{Сортировка пузырьком}

{Сортировка строк массива по убыванию минимальных элементов

в строке}

j:=1;

**while**(j<=str-1) **do**

**begin**

i:=1;

**while**(i<=str-j) **do**

**begin**

**if** Arr\_opening[i,col]<Arr\_opening[i+1,col] **then**

**begin**

k:=1;

**while** (k<=col) **do**

**begin**

temp:=Arr\_opening[i,k];

Arr\_opening[i,k]:=Arr\_opening[i+1,k];

Arr\_opening[i+1,k]:=temp;

k:=k+1;

**end**;

**end**;

i:=i+1;

**end**;

j:=j+1;

**end**;

{Вывод значений массива после второй сортировки}

i:=1;

**while**(i<=str) **do**

**begin**

j:=1;

**while**(j<=col) **do**

**begin**

write(Arr\_opening[i,j]:3);

write(' ');

**if** j=col-1 **then**

write(' | ');

j:=j+1;

**end**;

i:=i+1;

writeln;

**end**;

writeln('==================================');

readln

**End**.

Приложение Б

(обязательное)

Тестовые наборы

Тест 1 Тест 2

