Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

# оТЧЕТ

по лабораторной работе

на тему:

Анализ сортировок массивов

|  |  |  |
| --- | --- | --- |
| Выполнил  Студент гр. 951002 |  | В. Н. Протасеня |
| Проверил |  | Асс. Е.Е. Фадеева |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск, 2019

1. Теоретические сведения по теме лабораторной работы

**Массив** – упорядоченная совокупность однотипных элементов, имеющих

общее имя. Данное имя называется полной переменной , так как ее значением

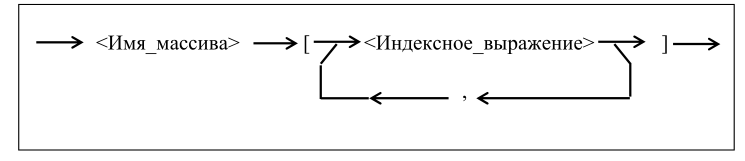
является весь массив.

**Тип элементов** массива называется базовым типом массива .

Для выделения отдельных элементов массива используются **индексы**

(индексные выражения). Индекс задает правило вычисления номера нужного

элемента массива.

****Для указания определенного элемента массива используется формат:

**Индексное выражение** – это выражение скалярного типа. Количество

индексных выражений определяется количеством измерений массива.

1.Задание на лабораторную работу

Анализ сортировок массивов (Сортировка пузырьком; Сортировка простыми вставками)

Произвести анализ сортировок массивов в соответствии с вариантом, выданным преподавателем. Для чего отсортировать по неубыванию массивы целочисленных элементов различной размерности: 10 элементов, 100 элементов, 2000 элементов. Анализ произвести по числу сравнений и числу обменов двух элементов. Исследования производить над массивами трех типов:

1. Массив, содержащий неотсортированные элементы
2. Массив, содержащий отсортированные элементы
3. Массив, содержащий элементы, отсортированные в обратном порядке

Все данные свести в таблицу.

* 1. Эскиз ожидаемого результата

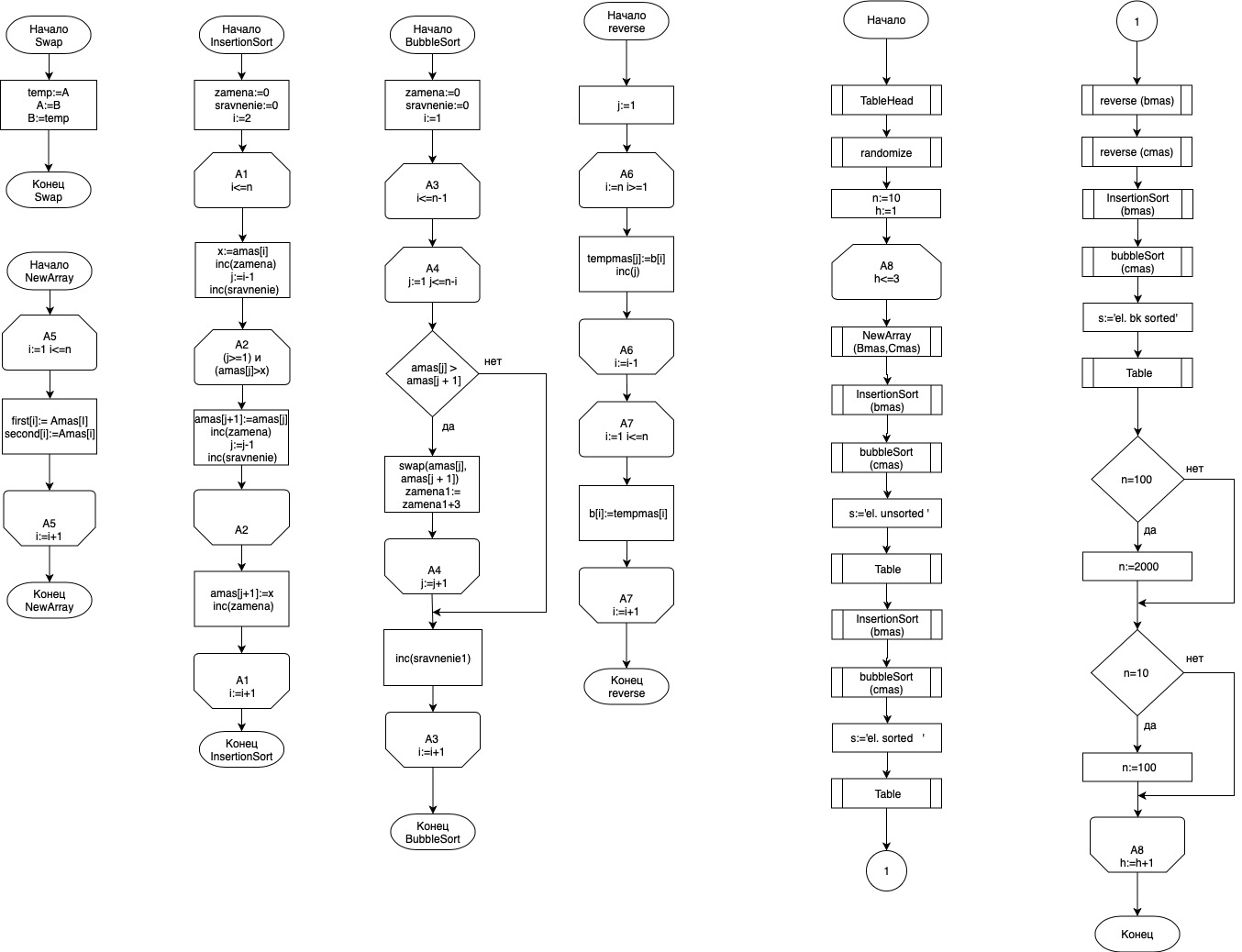
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Array**  **type** | **Bubble Sort #1** | | **Insertion Sort #2** | |
| **Number of comparisons** | **Number of exchanges** | **Number of comparisons** | **Number of exchanges** |
| **10**  **el. unsorted** |  |  |  |  |
| **10**  **el. sorted** |  |  |  |  |
| **10**  **el. bk sorted** |  |  |  |  |
| **100**  **el. unsorted** |  |  |  |  |
| **100**  **el. sorted** |  |  |  |  |
| **100**  **el. bk sorted** |  |  |  |  |
| **2000**  **el. unsorted** |  |  |  |  |
| **2000**  **el. sorted** |  |  |  |  |
| **2000**  **el. bk sorted** |  |  |  |  |

1. Выполнение
   1. Разработка алгоритма

Таблица 3.1 используемые идентификаторы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя идентификатора | назначение | Тип идентификатора | Начальное значение | Закон изменения | Имя цикла, в котором происходит изменение переменной |
| size | Размерность массива Amas | константа | 2000 | — | — |
| Cmas | Копия массива Amas для сортировки пузырьком | Tmas = array [1 .. size] of Integer | — | — | — |
| Amas | Начальный массив | Tmas = array [1 .. size] of Integer | — | — | — |
| Bmas | Копия массива Amas для сортировки простыми вставками | Tmas = array [1 .. size] of Integer | — | — | — |
| i | Параметр цикла | integer | 1, 2, n | i:=i+1  i:=i-1 | 51-74 |
| j | Параметр цикла | integer | 1 | j:=j+1 | 36-61 |
| sravnenie | Количество сравнений в сортировке простыми вставками | integer | 0 | inc(sravnenie) | 36-61 |
| zamena | Количество замен в сортировке простыми вставками | integer | 0 | inc(zamena) | 36-61 |
| sravnenie1 | Количество сравнений в сортировке пузырьком | integer | 0 | inc(sravnenie1) | 72 |
| zamena1 | Количество замен в сортировке пузырьком | integer | 0 | inc(zamena1)  zamena1 := zamena1 + 3 | 56-70 |
| n | Изменяющаяся размерность массивов bmas, cmas | integer | 10 | if n = 100 then  n := 2000;  if n = 10 then  n := 100 | 128-155 |
| h | Параметр цикла | integer | 1 | h:=h+1 | 128-155 |
| s | Переменна для отображения типа массива в таблице | string | — | s := 'el. unsorted '  s := 'el. sorted '  s := 'el. bk sorted' | 128-155 |

Рисунок 3.1 – Схема работы программы



3.2 Текст программы и его описание

program laba5;

{$APPTYPE CONSOLE}

uses

SysUtils;

const

size = 2000;

type

Tmas = array [1 .. size] of Integer;

var

Amas, Bmas, Cmas: Tmas;

i, j, sravnenie, zamena, zamena1, sravnenie1, n, h: Integer;

s: string;

procedure Swap(var A, B: Integer); // меняет местами элементы массива

var

temp: Integer;

begin

temp := A;

A := B;

B := temp;

end;

procedure TableHead;

begin

Writeln('---------------------------------------------------------------------------------');

Writeln('| | Bubble Sort #1 | Insertion Sort #2 |');

Writeln('| Array |---------------|---------------|---------------|----------------|');

Writeln('| type | Number of | Number of | Number of | Number of |');

Writeln('| | comparisons | exchanges | comparisons | exchanges |');

Writeln('|---------------|---------------|---------------|---------------|----------------|');

end;

procedure Table;

begin

Writeln('|', n:5,

' | | | | |');

Writeln('| ', s, ' | ', sravnenie1:13, ' | ', zamena1:14, '|', sravnenie:15,

'| ', zamena:15, '|');

Writeln('|---------------|---------------|---------------|---------------|----------------|');

end;

procedure InsertionSort(var Amas: Tmas);

var

x: Integer;

begin

zamena := 0;

sravnenie := 0;

for i := 2 to n do

begin

x := Amas[i];

inc(zamena);

j := i - 1;

inc(sravnenie);

while (j >= 1) and (Amas[j] > x) do

begin

Amas[j + 1] := Amas[j];

inc(zamena);

j := j - 1;

inc(sravnenie);

end;

Amas[j + 1] := x;

inc(zamena);

end;

end;

procedure BubbleSort(var Amas: Tmas);

begin

zamena1 := 0;

sravnenie1 := 0;

for i := 1 to n - 1 do

begin

for j := 1 to n - i do

begin

if Amas[j] > Amas[j + 1] then

begin

Swap(Amas[j], Amas[j + 1]);

zamena1 := zamena1 + 3;

end;

inc(sravnenie1);

end;

end;

end;

procedure NewArray(var first, second: Tmas); // копирует аmas в bmas cmas

begin

for i := 1 to n do

begin

first[i] := Amas[i];

second[i] := Amas[i];

end;

end;

procedure reverse(var B: Tmas); // разворот массивов

var

tempmas: Tmas;

begin

j := 1;

for i := n downto 1 do

begin

tempmas[j] := B[i];

inc(j);

end;

for i := 1 to n do

begin

B[i] := tempmas[i];

end;

end;

begin

TableHead;

randomize;

for i := 1 to size do // заполнение массива начального

begin

Amas[i] := random(100) - 50;

end;

n := 10;

for h := 1 to 3 do

begin

NewArray(Bmas, Cmas); // создание массивов amas bmas

InsertionSort(Bmas); // сортировки выбором для несортированного

BubbleSort(Cmas); // сортировка пузырьком

s := 'el. unsorted ';

Table;

InsertionSort(Bmas);

BubbleSort(Cmas);

s := 'el. sorted '; // для отсортированного

Table;

reverse(Bmas); // разворот массивов

reverse(Cmas);

InsertionSort(Bmas); // сортировка для обратно отсортированных массивов

BubbleSort(Cmas);

s := 'el. bk sorted';

Table;

if n = 100 then

n := 2000;

if n = 10 then

n := 100;

end;

readln;

end.

Описание программы: программа сортирует массивы, состоящие из 10, 100 и 2000 элементов. Используются сортировка пузырьком и простыми вставками.

* 1. Тестирование и отладка программы

Таблица 3.2 Прохождение тестов программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
|  | 1 |  |  | Тест пройден |
|  | 2 |  |  | Тест пройден |
|  | 3 |  |  |  |

* 1. Анализ прохождения тестов и полученных результатов

Программа прошла тесты успешно и ошибок не обнаружено.

* 1. Итоговый текст программы

{program laba5;

{$APPTYPE CONSOLE}

uses

SysUtils;

const

size = 2000;

type

Tmas = array [1 .. size] of Integer;

var

Amas, Bmas, Cmas: Tmas;

i, j, sravnenie, zamena, zamena1, sravnenie1, n, h: Integer;

s: string;

procedure Swap(var A, B: Integer); // меняет местами элементы массива

var

temp: Integer;

begin

temp := A;

A := B;

B := temp;

end;

procedure TableHead;

begin

Writeln('---------------------------------------------------------------------------------');

Writeln('| | Bubble Sort #1 | Insertion Sort #2 |');

Writeln('| Array |---------------|---------------|---------------|----------------|');

Writeln('| type | Number of | Number of | Number of | Number of |');

Writeln('| | comparisons | exchanges | comparisons | exchanges |');

Writeln('|---------------|---------------|---------------|---------------|----------------|');

end;

procedure Table;

begin

Writeln('|', n:5,

' | | | | |');

Writeln('| ', s, ' | ', sravnenie1:13, ' | ', zamena1:14, '|', sravnenie:15,

'| ', zamena:15, '|');

Writeln('|---------------|---------------|---------------|---------------|----------------|');

end;

procedure InsertionSort(var Amas: Tmas);

var

x: Integer;

begin

zamena := 0;

sravnenie := 0;

for i := 2 to n do

begin

x := Amas[i];

inc(zamena);

j := i - 1;

inc(sravnenie);

while (j >= 1) and (Amas[j] > x) do

begin

Amas[j + 1] := Amas[j];

inc(zamena);

j := j - 1;

inc(sravnenie);

end;

Amas[j + 1] := x;

inc(zamena);

end;

end;

procedure BubbleSort(var Amas: Tmas);

begin

zamena1 := 0;

sravnenie1 := 0;

for i := 1 to n - 1 do

begin

for j := 1 to n - i do

begin

if Amas[j] > Amas[j + 1] then

begin

Swap(Amas[j], Amas[j + 1]);

zamena1 := zamena1 + 3;

end;

inc(sravnenie1);

end;

end;

end;

procedure NewArray(var first, second: Tmas); // копирует аmas в bmas cmas

begin

for i := 1 to n do

begin

first[i] := Amas[i];

second[i] := Amas[i];

end;

end;

procedure reverse(var B: Tmas); // разворот массивов

var

tempmas: Tmas;

begin

j := 1;

for i := n downto 1 do

begin

tempmas[j] := B[i];

inc(j);

end;

for i := 1 to n do

begin

B[i] := tempmas[i];

end;

end;

begin

TableHead;

randomize;

for i := 1 to size do // заполнение массива начального

begin

Amas[i] := random(100) - 50;

end;

n := 10;

for h := 1 to 3 do

begin

NewArray(Bmas, Cmas); // создание массивов amas bmas

InsertionSort(Bmas); // сортировки выбором для несортированного

BubbleSort(Cmas); // сортировка пузырьком

s := 'el. unsorted ';

Table;

InsertionSort(Bmas);

BubbleSort(Cmas);

s := 'el. sorted '; // для отсортированного

Table;

reverse(Bmas); // разворот массивов

reverse(Cmas);

InsertionSort(Bmas); // сортировка для обратно отсортированных массивов

BubbleSort(Cmas);

s := 'el. bk sorted';

Table;

if n = 100 then

n := 2000;

if n = 10 then

n := 100;

end;

readln;

end.