Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

«Уральский федеральный университет имени первого Президента России Б.Н Ельцина»

ИРИТ-РТФ

Кафедра информационных технологий

О Т Ч Е Т

По Алгоритмизации и структурам данных

Лабораторная работа №3

Студент Трефилов А.А.

Группа РИ-240002

Преподаватель Спиричева Н.Р.

Екатеринбург 2015

Задание: Написать 2 метода сортировки: Шейкером и Деревом.

Код:

typedef struct tree

{

int b; // данные

struct tree \*left; // левый сын

struct tree \*right; // правый сын

} TREE;

TREE \*add\_to\_tree(TREE \*root, int new\_value)

{

if (root==NULL) // если нет сыновей - создаем новый элемент

{

root = (TREE\*)malloc(sizeof(TREE));

root->b = new\_value;

root->left = root->right = 0;

return root;

}

if (root->b < new\_value) // добавлем ветвь

{root->right = add\_to\_tree(root->right, new\_value);

}

else

root->left = add\_to\_tree(root->left, new\_value);

return root;

}

void tree\_to\_array(TREE \*root, int b[]) // процедура заполнения массива

{

static int max2=0; // счетчик элементов нового массива

if (root==NULL) return; // условие окончания - нет сыновей

tree\_to\_array(root->left, b); // обход левого поддерева

b[max2++] = root->b;

tree\_to\_array(root->right, b); // обход правого поддерева

free(root);

}

void SortTree(int b[], int elem\_total) // собственно сортировка

{

TREE \*root;

int i;

int schet=0;

root = NULL;

for (i=0; i<elem\_total; i++) // проход массива и заполнение дерева

{

root = add\_to\_tree(root, b[i]);

schet ++;

}

tree\_to\_array(root, b); // заполнение массива

cout << "Сортировка деревом: " << endl;

for (int i=0;i<15;i++)

{cout << b[i] << " ";}

cout << endl << "Количество действий: " << schet;

cout << endl << endl;

schet=0;

}

void SheikerSort(int \*b, const int n)

{

int schet=0;

int l, r, i, k, buf;

k = l = 0;

r = n - 2;

while(l <= r)

{

for(i = l; i <= r; i++)

if (b[i] > b[i+1])

{

buf = b[i]; b[i] = b[i+1]; b[i+1] = buf;

k = i;

}

r = k - 1;

for(i = r; i >= l; i--)

if (b[i] > b[i+1])

{

buf = b[i]; b[i] = b[i+1]; b[i+1] = buf;

k = i;

}

l = k + 1;

schet ++;

}

cout << "Шейкерная сортировка: " << endl;

for(int i = 0; i < 15; i++)

{ cout << b[i] << " ";}

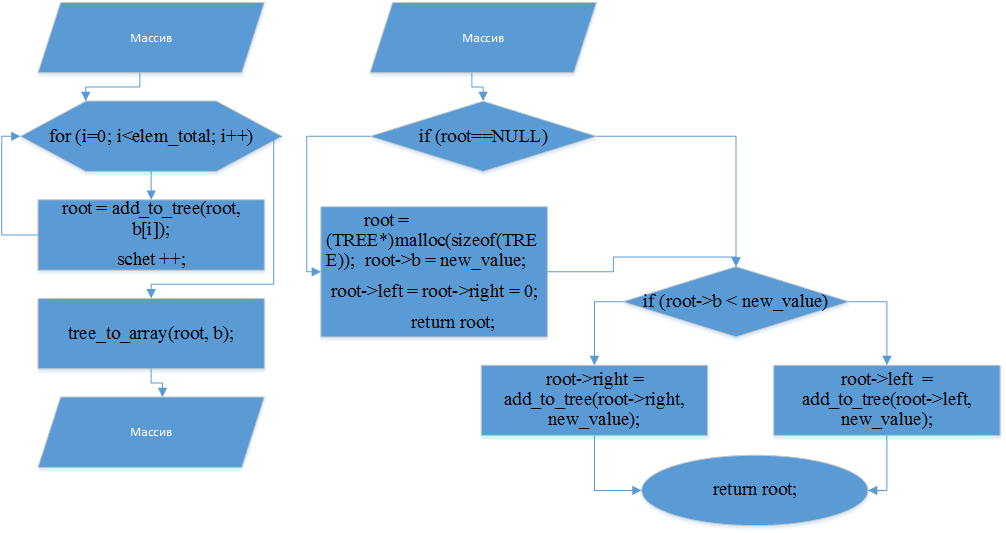
cout << endl << "Количество действий: " << schet;

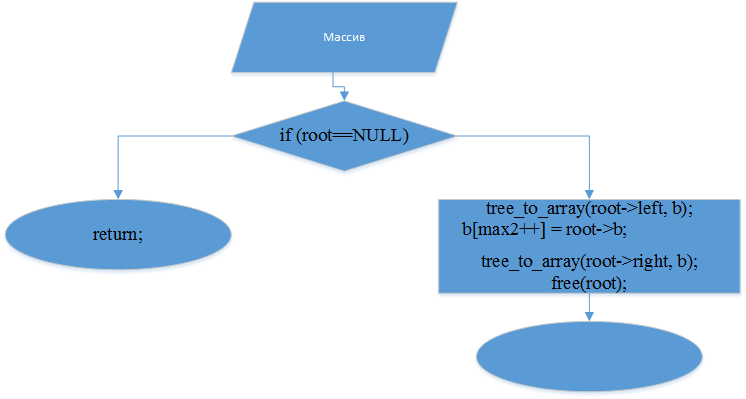
cout << endl << endl;

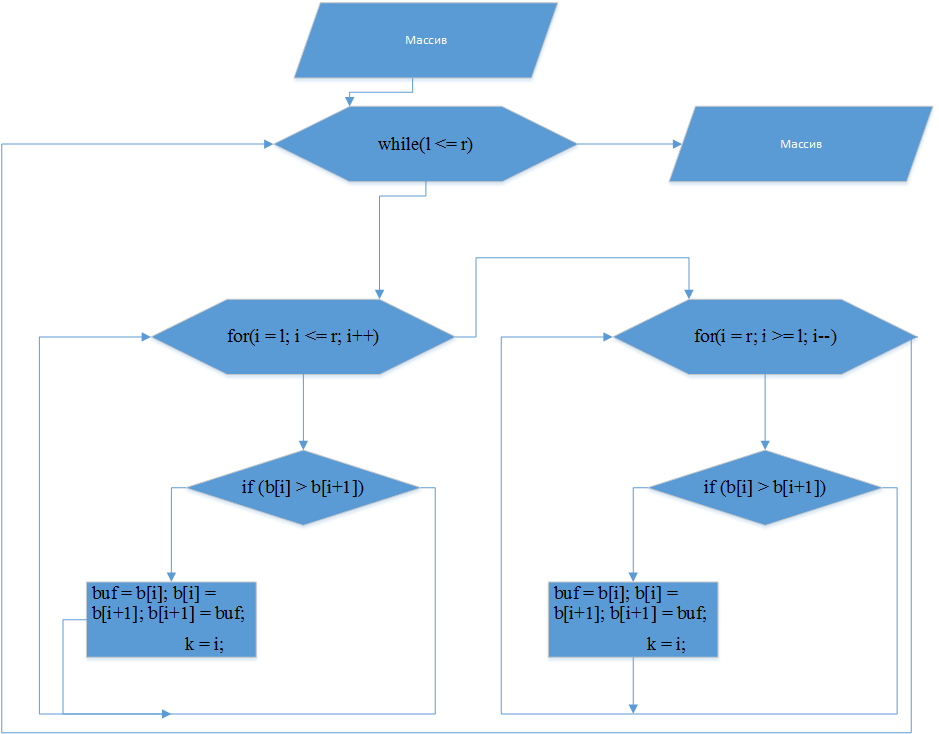
schet=0;

}

Блок- схема







Вывод: в данной лабораторной работе я сравнил два вида сортировки при работе с тремя массивами: прямой(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15), обратный(15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,) и случайный(….). Благодаря счетчику выяснилось, что сортировка Деревом во всех случаях одинакова по количеству действий, так как она создает новый массив из всех данных чисел. А сортировка Шейкером достаточно хороша, так как для сортировки массива из 15 чисел ей понадобиться сравнить от 1 до 7(половина массива) раз.