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

ОТЧЕТ

по лабораторной работе

на тему «Быстрая сортировка по классу медианы из трёх»

Выполнил студент гр. Б8217

Некрасов Евгений Андреевич

Проверила:

Крестникова О.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Владивосток

2019

1.Неформальная постановка задачи:

1.1 Быструю сортировку по классу медианы из трёх.

Примечание: для упрощения реализации воспользуйтесь шаблоном векторов заполнив его целочисленными значениями.

1.2 Реализовать методы для работы сортировки :

Создание вектора, заполнение вектора, печать вектора.

Быструю сортировку, метод обмена swap, выборку медианы из трёх.

2. Формальная постановка задачи:

Область исходных данных:

X-множество входных данных:

int printVec (std::vector <int> &vec) {

std::cout << "Печать вектора" << std::endl;

for (int i = 0; i < vec. size (); i++) {

std::cout << vec [i] << std::endl;

}

return 0;

}

int swap (int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

return 0;

}

int quitSort (std::vector <int> &vec, int first, int last) {

int i = first;

int j = last;

int x = vec [(first + last) / 2];

do {

while (vec [i] < x) i++;

while (vec [j] > x) j--;

if (i <= j) {

if (vec [i] > vec [j]) swap (&vec [i], &vec [j]);

i++;

j--;

}

}

while (i <= j);

if (i < last) {

quitSort (vec, i, last);

}

if (first < j) {

quitSort (vec, first, j);

}

return 0;

}

std::vector <int> createVector () {

std::vector <int> vectorData;

std::cout << "Создайте последовательность хаотичного типа" << std::endl;

std::cout << "Для выхода из цикла введите ноль" << std::endl;

int readData;

for (;;) {

std::cin >> readData;

if (readData == 0) {

std::cout << "Вы завершили ввод" << std::endl;

break;

}

std::cout << "Вы ввели число: " << readData << std::endl;

vectorData. push\_back (readData);

}

return vectorData;

}

int exchange (std::vector <int> &vec, int i, int j) {

int k;

k = vec [i];

vec [i] = vec [j];

vec [j] = k;

return 0;

}

int mediana (std::vector <int> &vec) {

int size = vec. size ();

int i = size / 4;

if (vec [i] > vec [i \* 2]) {

if (vec [i] > vec [i \* 3]) {

exchange (vec, i, size);

}

else {

exchange (vec, i \* 3, size);

}

}

else {

if (vec [i \* 2] > vec [i \* 3]) {

exchange (vec, i \* 2, size);

}

}

quitSort (vec, 0, vec. size () - 1);

return 0;

}

3. Спецификация данных:

INPUT:

Данные считываются из потока ввода std::in;.

Формат входного значения: int

Данные выводятся в поток вывода std::out

Формат выходных данных: int

4. Тесты

Вывод из консоли перенаправлен в файл командой

"quicSort | tee conclusion.txt"

Создайте последовательность хаотичного типа

Для выхода из цикла введите ноль

Вы ввели число: 7

Вы ввели число: 1

Вы ввели число: 3

Вы ввели число: 2

Вы ввели число: 6

Вы ввели число: 5

Вы ввели число: 11

Вы ввели число: 33

Вы ввели число: 21

Вы ввели число: 27

Вы ввели число: 26

Вы завершили ввод

Исходный вектор

Печать вектора

7

1

3

2

6

5

11

33

21

27

26

Результирующий вектор

Печать вектора

1

2

3

5

6

7

11

21

26

27

33