Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Выполнила студентка группы КС-30 (Ноль Эльвира Гарриевна)

Ссылка на репозиторий: (https://github.com/Elviranng/Nol-Elvira-KS-30/blob/master/lab1.sem2)

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: (05.06.2021)

Оглавление

[Описание задачи. 2](#_Toc70459347)

[Описание метода/модели. 3](#_Toc70459348)

[Выполнение задачи. 4](#_Toc70459349)

[Заключение. 8](#_Toc70459350)

# Описание задачи.

В рамках лабораторной работы необходимо изучить и реализовать метод сортировки соответствующий вашему варианту (приведены ниже).  
Для реализованного метода сортировки необходимо провести серию тестов для всех значений N из списка (1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000), при этом:

* в каждом тесте необходимо по 20 раз генерировать вектор, состоящий из N элементов
* каждый элемент массива заполняется случайным числом с плавающей запятой от -1 до 1, для этого можно использовать как C функцию rand(), так и С++ генераторы:

mt19937 engine(time(0));

uniform\_real\_distribution<double> gen(-1.0, 1.0);

for(auto& el: v)

el = gen(engine);

* каждый массив после генерации необходимо отсортировать и замерить время, требуемое на сортировку, для замера времени использовать следующий код:

#include <chrono>

int main(){

…

chrono::high\_resolution\_clock::time\_point start = chrono::high\_resolution\_clock::now();

sort(v);

chrono::high\_resolution\_clock::time\_point end = chrono::high\_resolution\_clock::now();

chrono::duration<double, nano> nano\_diff = end - start;

chrono::duration<double, micro> micro\_diff = end - start;

chrono::duration<double, milli> milli\_diff = end - start;

chrono::duration<double> sec\_diff = end - start;

cout << endl;

cout << "Time: " << nano\_diff.count() << " nano sec." << endl;

cout << "Time: " << micro\_diff.count() << " micro sec." << endl;

cout << "Time: " << milli\_diff.count() << " milli sec." << endl;

cout << "Time: " << sec\_diff.count() << " sec." << endl;

}

* Результат замера для каждой попытки каждого теста записать в файл общий файл.

По окончанию всех тестов необходимо нанести все точки, полученные в результате замеров времени на график где на ось абсцисс(Х) нанести N, а на ось ординат(Y) нанести значения времени на сортировку. По полученным точкам построить график лучшего (минимальное время для каждого N), худшего (максимальное время для каждого N) и среднего (среднее время для каждого N) случая.

# Описание метода/модели.

Сортировка вставками (англ. Insertion sort) — алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов

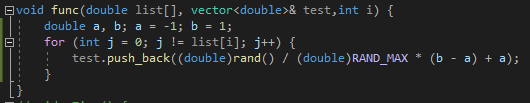
На вход алгоритма подаётся последовательность чисел: a1, a2,..., an. Сортируемые числа также называют ключами. Входная последовательность на практике представляется в виде массива с n элементами. На выходе алгоритм должен вернуть перестановку исходной последовательности a1, a2,..., an, чтобы выполнялось следующее соотношение:  
a1 ≤ a2 ≤ … ≤ an

В начальный момент отсортированная последовательность пуста. На каждом шаге алгоритма выбирается один из элементов входных данных и помещается на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. В любой момент времени в отсортированной последовательности элементы удовлетворяют требованиям к выходным данным алгоритма.

Данный алгоритм можно ускорить при помощи использования бинарного поиска для нахождения места текущему элементу в отсортированной части. Проблема с долгим сдвигом массива вправо решается при помощи смены указателей.

# Выполнение задачи.

1. Функция заполнения вектора:



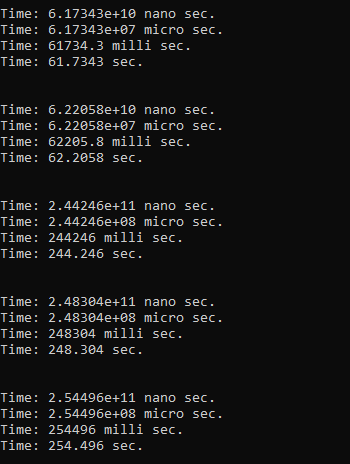
Здесь происходит заполнение вектора случайными числами.

1. main:



В main происходит сортировка заданным методом, считается и выводится время.

1. Результат выполнения программы

# Заключение.

Сортировка вставками это простой алгоритм, часто используется при изучении компьютерных наук и обычно, является одним из первых алгоритмов сортировки, с которым знакомят учеников. Он интуитивно понятен и прост в реализации, но он очень медленный для больших массивов и поэтому почти никогда не используется для сортировки реальных данных.