**Задание на лабораторную #1.**

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.
* При замере времени выбираем только один из варивантов (nano\_diff, micro\_diff, milli\_diff, sec\_diff).

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.

В качестве дополнительного задания, необходимо построить график худшего случая, и график O(c \* g(N)), где g(N) соответствует асимптотической сложности рассматриваемого метода сортировки, подобрав такое значение C, что бы начиная с N ~ 1000 график асимптотической сложности возрастал быстрее чем полученное худшее время, но при этом был различим на графике.

Варианты сортировок:

1. Сортировка пузырьком
2. Сортировка перемешиванием
3. Сортировка вставками
4. Сортировка выбором

Теоретическое описание сортировок с их асимтотической сложностью приведены на [сайте](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8).

Сдача лабораторной работы осуществляется посредством подготовки отчета в формате приведенном по [ссылке](https://github.com/MUCTR-IKT-CPP/algorithms-2cpp/blob/master/%D0%9C%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B/%D0%A9%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%B0.docx)