###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«ИЗУЧЕНИЕ ОПТИМИЗИРУЮЩЕГО КОМПИЛЯТОРА»

студента (ки) 2 курса, 22202 группы

**Ганькеева Максима Сергеевича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

(ученая степень, звание)

В. А. Перепёлкин

Новосибирск 2023

# ЦЕЛЬ

1. Изучение основных функций оптимизирующего компилятора, и некоторых примеров оптимизирующих преобразований и уровней оптимизации.

2. Получение базовых навыков работы с компилятором GCC.

3. Исследование влияния оптимизационных настроек компилятора GCC на время исполнения программы.

# ЗАДАНИЕ

Алгоритм сортировки методом пузырька. Дан массив случайных чисел длины N.

# ОПИСАНИЕ РАБОТЫ

График зависимости времени работы программы от уровня оптимизации и выбранного числа элементов N в программе:

Листинг программы:

1. #include <stdio.h>
2. #include <time.h>
3. #include <stdlib.h>
4. #define SIZE 140000
5. **void** swap(**int** \*a, **int** \*b) {
6. **int** tmp = \*a;
7. \*a = \*b;
8. \*b = tmp;
9. }
10. **int**\* FillArray(**long** **long** n) {
11. **int** \*array = calloc(n, **sizeof**(**int**));
12. **for** (**int** i = 0; i < n; i++) {
13. array[i] = rand() % 1000;
14. }
15. **return** array;
16. }
17. **int** main() {
18. **struct** timespec start, end;
19. srand(time(NULL));
20. **long** **long** n = SIZE;
21. **int** \*array = FillArray(n);
22. clock\_gettime(CLOCK\_MONOTONIC\_RAW, &start);
23. **for** (**int** i = 0; i < n; i++) {
24. **for** (**int** j = 0; j < n - 1; j++) {
25. **if** (array[j] > array[j + 1]) {
26. swap(&array[j], &array[j + 1]);
27. }
28. }
29. }
30. clock\_gettime(CLOCK\_MONOTONIC\_RAW, &end);
31. printf("Time taken: %lf sec.\n", end.tv\_sec-start.tv\_sec + 0.000000001 \* (end.tv\_nsec-start.tv\_nsec));
32. **return** 0;
33. }

Команды для компиляции:

* gcc -O0 main.c -o main.bin
* gcc -O1 main.c -o main.bin
* gcc -O2 main.c -o main.bin
* gcc -O3 main.c -o main.bin
* gcc -Os main.c -o main.bin
* gcc -Ofast main.c -o main.bin
* gcc -Og main.c -o main.bin

# ЗАКЛЮЧЕНИЕ

В процессе выполнения лабораторной работы я изучил основные функции и характеристики компилятора, основные уровни оптимизации компилятора GCC и ключи их запуска. Получили базовые навыки работы с компилятором GCC, исследовали влияние оптимизационных настроек компилятора GCC на время исполнения программы.

По результатам нетрудно заметить, что самые лучшие временные результаты были получены при использовании уровней оптимизации O1 и O2.