

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №1**  
**по курсу «Программирование графических процессоров»**

*Освоение программного обеспечения для работы с технологией CUDA.*

*Примитивные операции над векторами.*

Выполнил: Попов М. Р.

Группа: 8О-408Б

Преподаватели: К.Г. Крашенинников,  
А.Ю. Морозов

Москва, 2023

## Условие

1. **Цель работы:** Ознакомление и установка программного обеспечения для работы с программно-аппаратной архитектурой параллельных вычислений (CUDA). Реализация одной из примитивных операций над векторами.
2. **Вариант 4:** Поэлементное нахождение минимума векторов

## Программное и аппаратное обеспечение

1. Графический процессор: Nvidia GeForce GT 545
  - a. Количество потоковых процессоров: 144
  - b. Частота ядра: 720 МГц
  - c. Количество транзисторов: 1.170 млн
  - d. Тех. процесс: 40 нм
  - e. Энергопотребление: 70 Вт
2. ОС: Ubuntu 16.04
3. Текстовый редактор: VS Code
4. Компилятор: nvcc

## Метод решения

Пройдёмся в цикле по каждому элементу из обоих векторов и сравним их поэлементно, наименьший положим в результирующий вектор по соответствующему индексу.

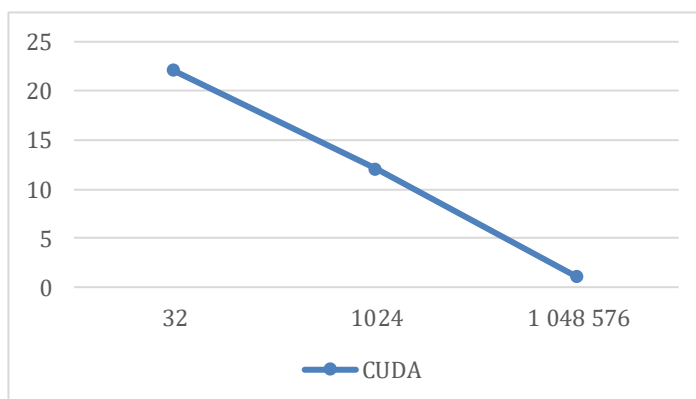
## Описание программы

Программа состоит из одного файла, в котором есть функция **kernel**, внутри которой и выполняется проход в цикле по векторам, а также функция **get\_min**, которая возвращает минимальное из двух чисел с плавающей точкой.

## Результаты

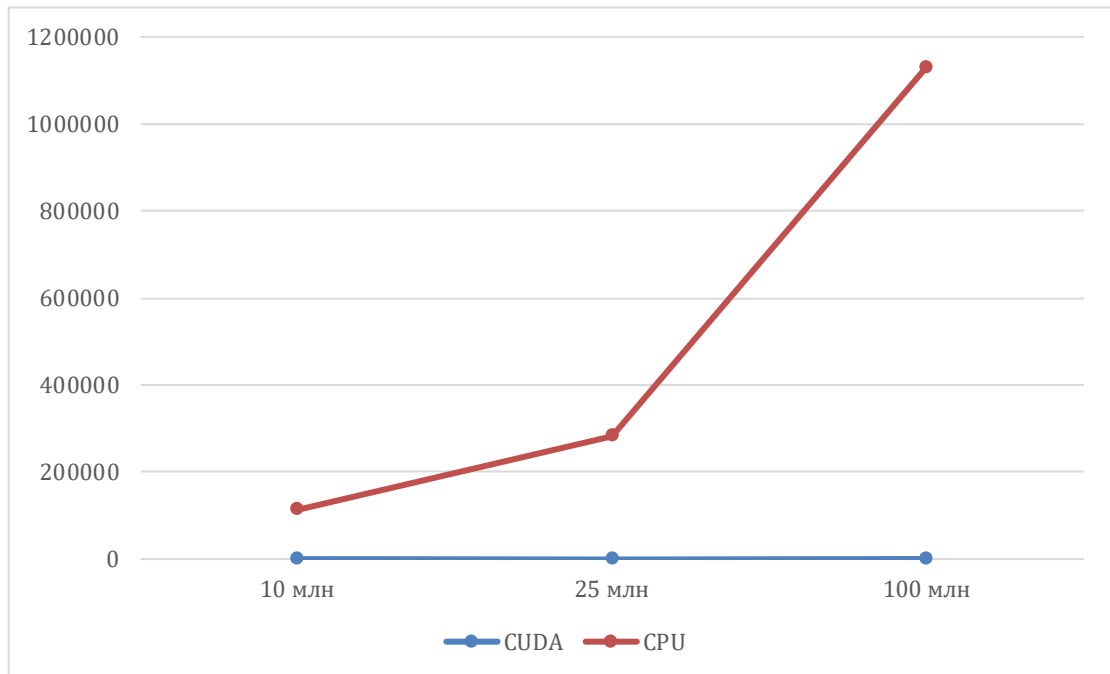
1. Зависимость времени выполнения программы от количества используемых потоков (для тестов использовались два вектора по 10 миллионов чисел):

Потоки	Время (в мс)
1×32	22
32×32	12
1024×1024	1



2. Сравнение программы на CUDA с  $32 \times 32$  потоками и программы на CPU с одним потоком:

Размер векторов	Время на CUDA (в мс)	Время на CPU (в мс)
10 млн	12	113331
25 млн	24	283675
100 млн	89	1131260



## Выводы

Проделав лабораторную работу, я использовал базовые операции для работы с CUDA, чтобы реализовать алгоритм нахождения поэлементного минимума двух векторов. Также, проведя тестирование своей программы, я убедился, что видеокарта намного эффективнее процессора в многопоточных вычислениях.