

# Результаты эксперимента

Артем Демченко, Александра Лановая, Артем Ржанков

# Эксперимент

**Цель:** Сравнение средней работы различных реализаций алгоритмов Боруки и Прима

**Шаги:**

- Провести 20 запусков алгоритмов
- Построить доверительные интервалы
- Проанализировать результаты

**Вопросы:**

- На какой из трех библиотек (**spla**, **gunrock**, **graphblas(lagraph)**) быстрее реализация алгоритма Боруки?
- На какой из двух библиотек (**spla** и **gunrock**) быстрее реализация алгоритма Прима?

# Характеристики машины

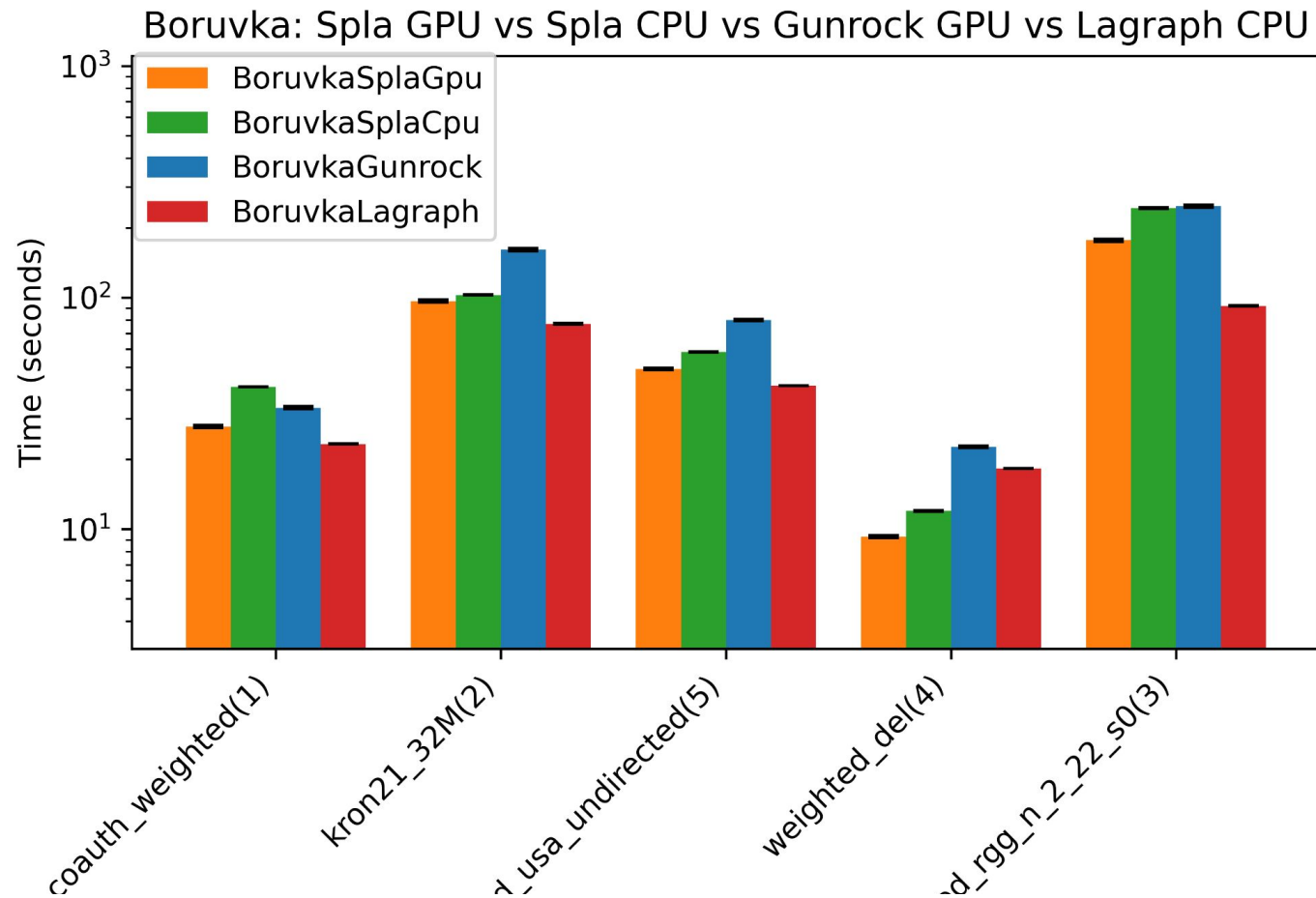
- ОС Ubuntu 24.04 LTS
- Видеокарта NVIDIA GeForce RTX 3050 Ti Laptop GPU
  - 4Gb VRAM
  - Драйвер версии 32.0.15.5597
  - CUDA Toolkit 12.3
  - 2560 CUDA-ядер
- Процессор 11th Gen Intel Core i7-11800H @ 2.30GHz
  - Кэш L1 – 640Kb, Кэш L2 – 10Mb, Кэш L3 – 24Mb
- 16 GB RAM
- OpenCL:
  - Версия: OpenCL 3.0
  - [PoCL 7.0](#)

# Датасет

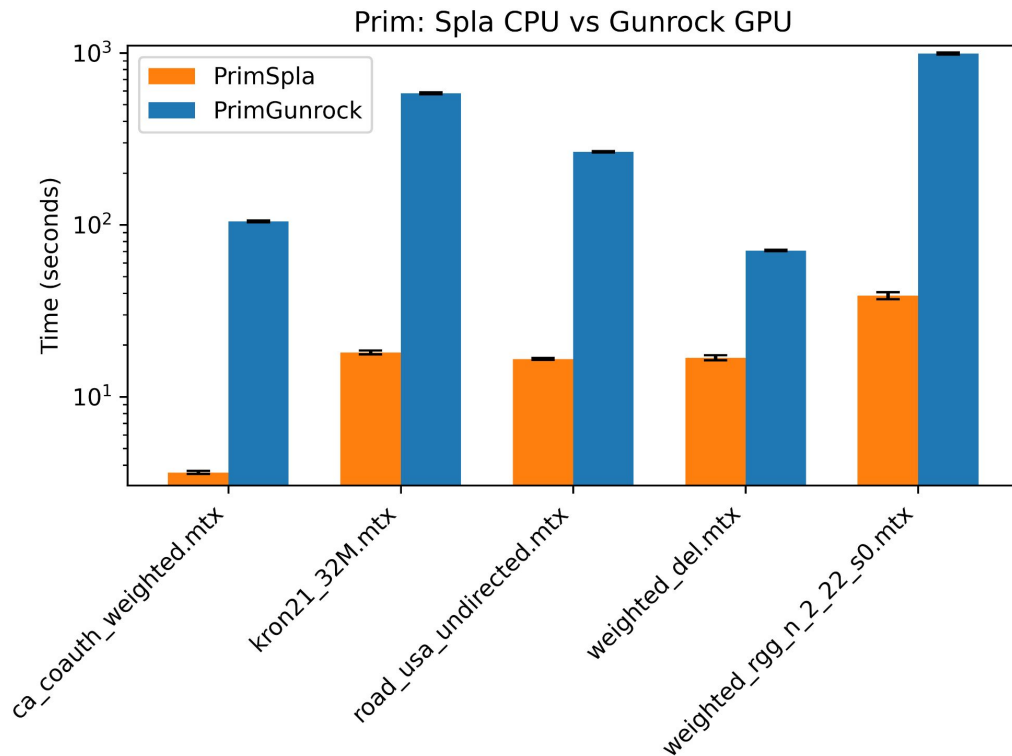
Граф	Количество вершин, $\times 10^6$	Количество ребер, $\times 10^6$
CA-Coauthors-Dblp (1)	0.54	15
Graph500 Kronecker (R-MAT) (2)	2	32
RGG (3)	4.1	30
Delaunay-n24 (4)	4.1	3.1
USA-road-d.USA (5)	4.1	8.5

[“A High-Performance MST Implementation for GPUs” 2023](#)

На какой из двух библиотек быстрее реализация алгоритма Борувки?



На какой из двух библиотек быстрее реализация алгоритма Прима?



# Результаты

1. **Борувка** на всех тестовых графах отрабатывает быстрее в Spla, чем в Gunrock
2. **Прим** во всех случаях быстрее в Spla на CPU, чем в Gunrock на GPU: разница в среднем – почти на порядок в пользу Spla на CPU