



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Борувка. SPLA и Pregel+. Презентация 1.

Пилюк Дмитрий Алексеевич

Санкт-Петербург  
2025

# Постановка задачи

Дан неориентированный взвешенный граф  $G = (V, E)$ . Требуется найти минимальное остовное дерево  $G$ .

## Определение

**Остовное дерево** — ациклический связный подграф данного связного неориентированного графа, в который входят все его вершины.

## Определение

**Минимальное остовное дерево** — остовное дерево данного графа минимального веса.

## Принцип алгоритма

- Каждая вершина отдельная компонента связности
- Для каждой компоненты определяем минимальное ребро, связывающее с другой компонентой. Объединяем компоненты этим ребром
- Повторяем пока есть ребра между разными компонентами

# Алгоритм Боруки в терминах линейной алгебры

- $S$  — матрица смежности графа
- $(w, i)$  — представление ребра, где  $w$  — вес ребра, а  $i$  — компонента с которой оно связывает
- Вектор  $\text{edge}$ , где  $\text{edge}[u]$  минимальное ребро инцидентное вершине  $u$
- Вектор  $\text{cedge}$ , где  $\text{cedge}[u]$  минимальное ребро инцидентное одному из потомков  $u$ , если  $u$  корень, в противном случае  $+\infty$
- Вектор  $\text{parent}$ , где  $\text{parent}[u]$  — номер компоненты(корень) для  $u$

# Алгоритм Борувки в терминах линейной алгебры

Полукольцо  $\text{combMin}$  над  $(\mathbb{R} \cup \{+\infty\}) \times V$ :

- "сложение" —  $\min((w_1, i_1), (w_2, i_2))$
- "умножение" —  $\text{comb}((w, i), j) = (w, j)$
- "0" —  $+\infty$

# Алгоритм Борувки в терминах линейной алгебры

while  $S \neq \emptyset$

- $edge = S \times parent$
- $cedge[parent[u]] = \min(cedge[parent[u]], edge[i])$
- актуализируем parent
- удаляем из  $S$  все ребра внутри компонент

# Алгоритм Борувки в терминах Pregel

- Каждая вершина содержит указатель на свою компоненту (supervertex). Изначально каждая вершина корень компоненты
- Локальный выбор: каждая вершина выбирает своё минимальное исходящее ребро и посылает в supervertex
- Supervertex выбирают глобально минимальное ребро  $(u, l, v)$ , запрашивает supervertex вершины  $v$
- Слияние компонент: добавляются ребра в  $mst$ , для каждой компоненты выбираем новый supervertex
- Протягивание указателей: делаем для каждой  $u$  пока  $supervertex(supervertex(u)) \neq supervertex(u)$
- Очистка рёбер: каждая вершина удаляет свои ребра в вершины этой же компоненты
- Если у вершины не осталось ребер, голосует за остановку

# Эксперимент

На какой из библиотек spla(CPU(POCL) и GPU(NVIDIA)) или Pregel+(разное количество Workers) алгоритм Боровки быстрее?

- 15 запусков алгоритма
- Составить информацию о среднем значении и доверительных интервалах
- Визуализировать информацию

Граф	V	E
USA-road-d.USA	23.9M	58.3M
USA-road-d.CTR	14.1M	34.3M
USA-road-d.W	6.3M	15.2M
USA-road-d.LKS	2.8M	6.9M
USA-road-d.FLA	1.1M	2.8M
USA-road-d.NY	264K	734K

# Характеристика вычислительной машины

- CPU: AMD Ryzen 7 5800H
  - ▶ 8 ядер
  - ▶ hyperthreading выключен
  - ▶ L1d: 256 KiB (8 instances)
  - ▶ L2: 4 MiB (8 instances)
  - ▶ L3: 16 MiB (1 instance)
- GPU: NVIDIA GeForce GTX 1650 (4096MiB)
- RAM: 16 GB
- OS: Ubuntu 25.10