Теория графов. Презентация 1

Ефремов Алексей, Сичкар Георгий, Кононова Юлия

Multiple-source Parent BFS

Поставновка задачи (MS-Parent BFS)

Задача

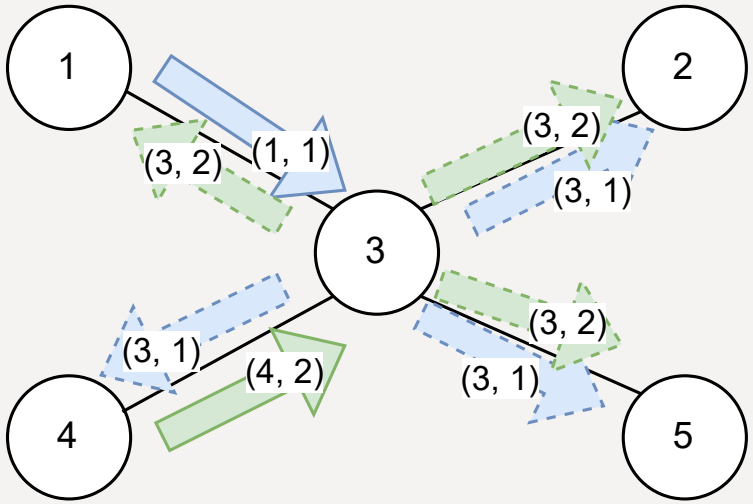
Дан неориентированный невзвешенный граф и заданное множество стартовых вершин S.

- Для каждой стартовой вершины $s \in S$ выполнить *независимый* Parent BFS. Для каждой пары вершин (s, v), где $s \in S$, определить родительскую вершину v на одном из кратчайших путей (из s в v)

(Анализ социальных сетей...)

MS-BFS Pregel

- Из всех стартовых вершин отправляем сообщение соседям `(номер_вершины, стартовая_группа)`
- При принятии сообщения, если в итоговой таблице нет записи:
  - Добавляем запись в таблицу
  - Распространяем сообщение (заменяя `номер_вершины`)



Тестовый датасет (социальные сети)

- SNAP: графы социальных сетей
- 2013: Direction-optimizing breadth-first search

Тестовый датасет (социальные сети)

Description	Nodes	Edges
(SNAP)		
Twitch gamers network	168,114	6,797,557
Gemsec Facebook dataset	134,833	1,380,293
Gemsec Deezer dataset	143,884	846,915
Twitch social networks	34,118	429,113
Github developer network	37,700	289,003
Facebook page-page network	22,470	171,002
Facebook social circles	4,039	88,234
Deezer Europe social network	28,281	92,752
LastFM Asia social network	7,624	27,806

Борувка

Поставновка задачи (Борувка)

Задача

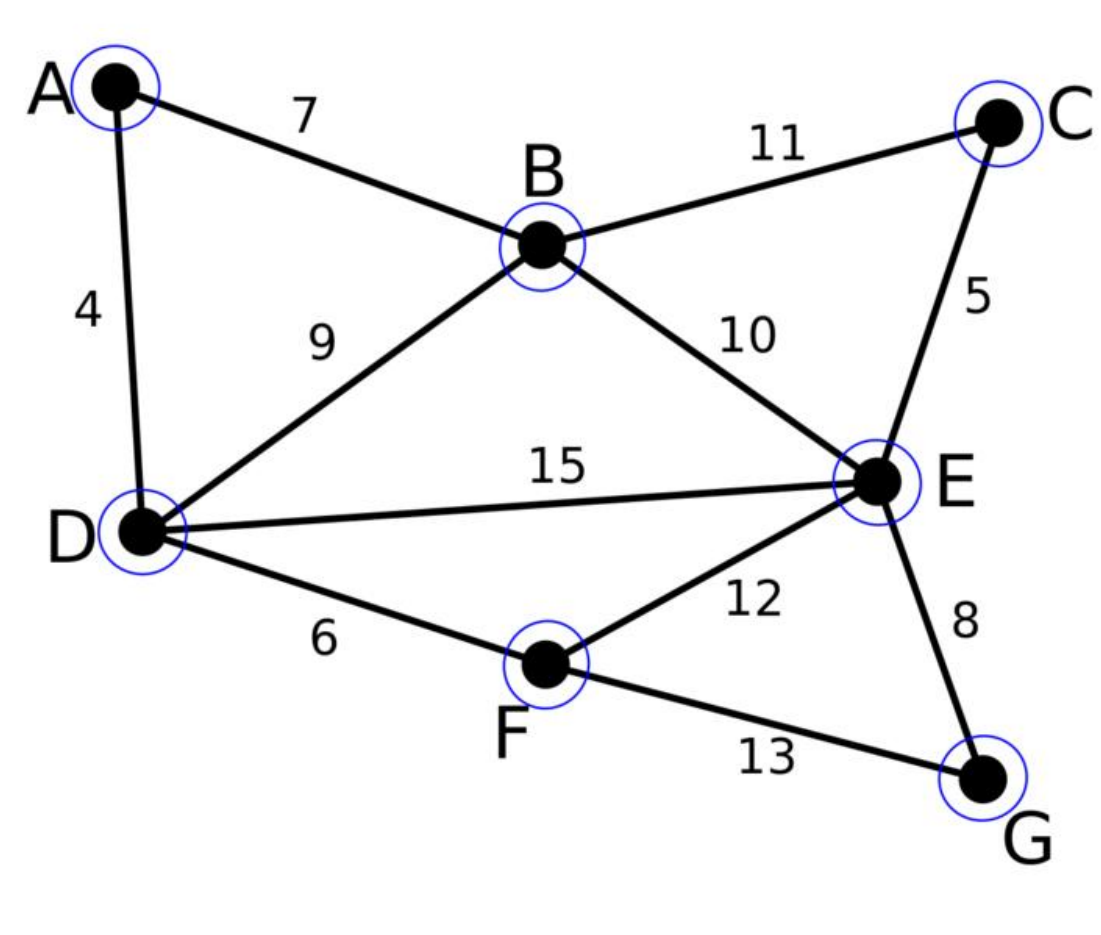
Есть **неориентированный взвешенный граф** G. Необходимо найти минимальное сотовное дерево для G.

Остовное дерево — ациклический связный подграф G, в который входят все вершины из G.

Минимальное остовное дерево (*MST*) — остовное дерево G минимального веса.

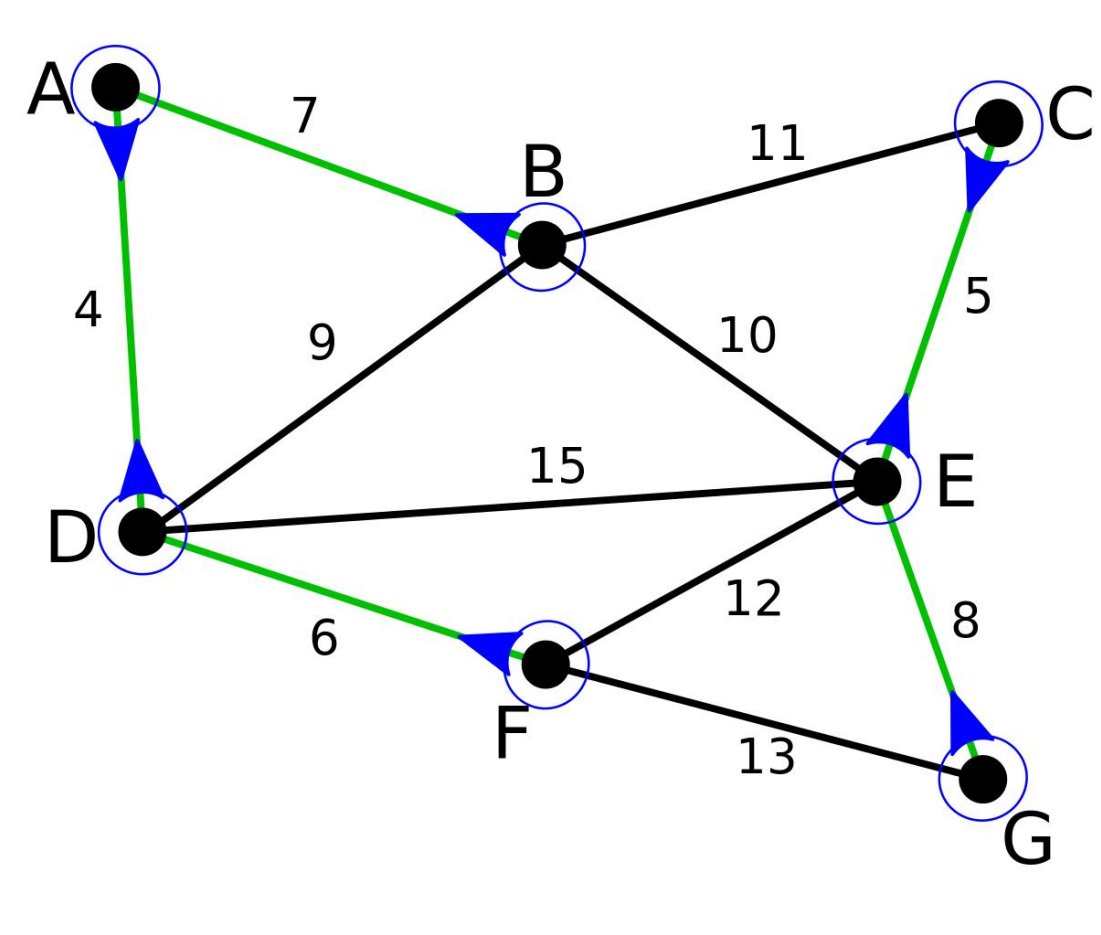
(Моделирование, оптимизации; подключение водопровода к каждому дому в районе...)

Борувка



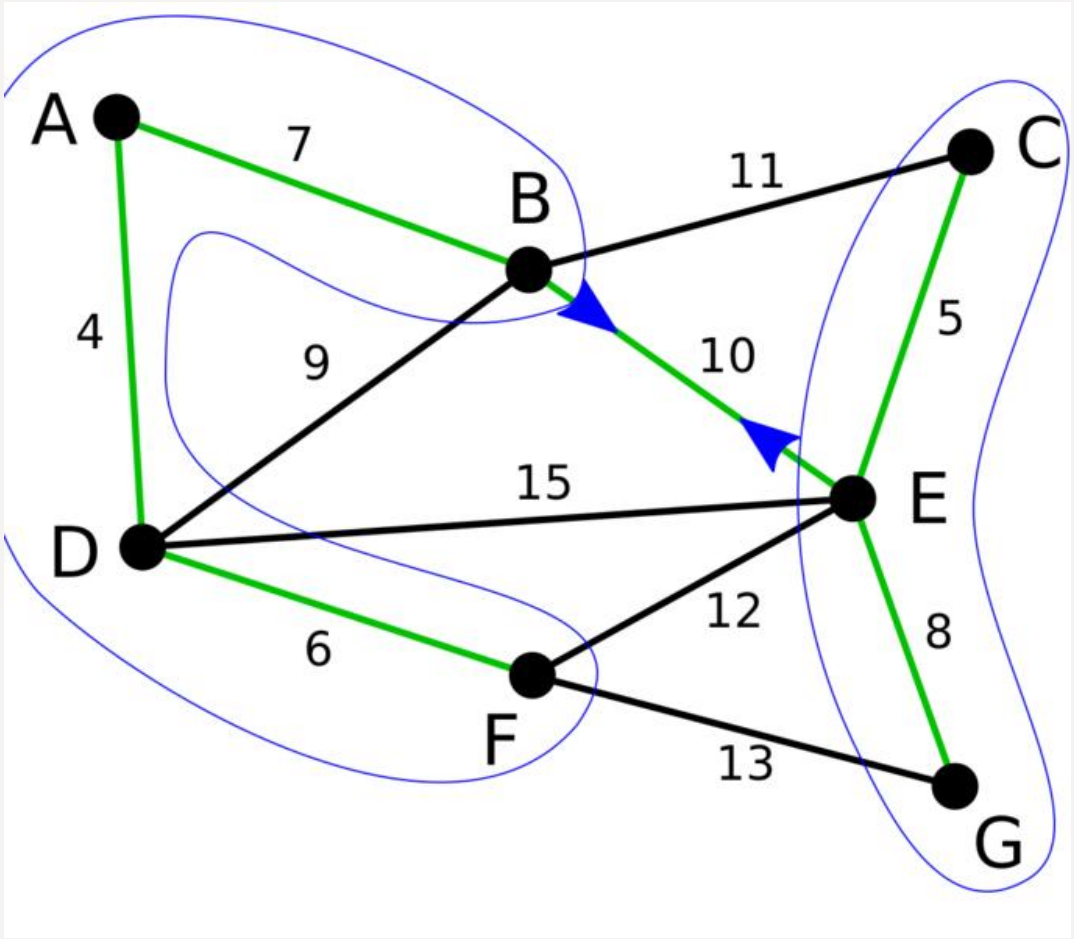
- {A}
- {B}
- {C}
- …

Борувка



- {ABDF}
- {CEG}

Борувка



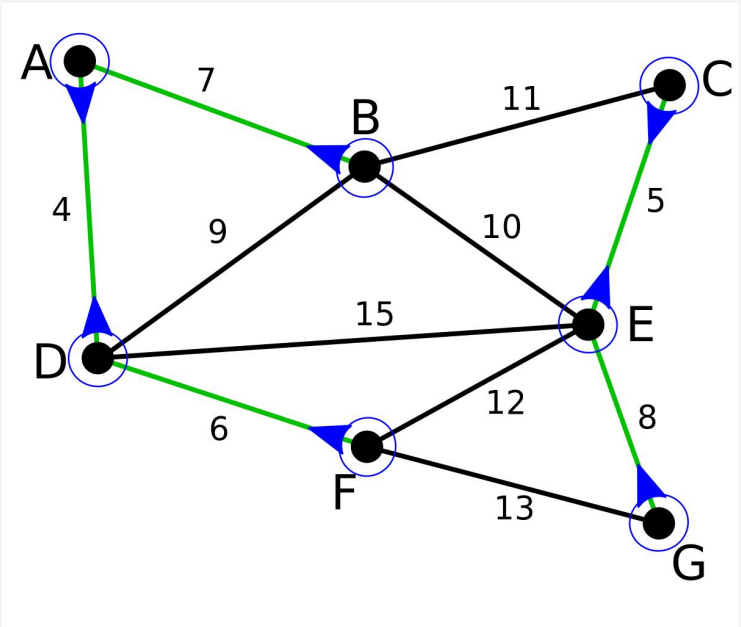
- {ABDFCEG}

Борувка Pregel

- Деревья == вершины
  - Итерация на множестве несвязанных вершин $\rightarrow$ множество деревьев
  - Перестройка множества деревьев $\rightarrow$ множество несвязанных вершин

Борувка Pregel: 4 шага

1. **Выбор минимального ребра**
   - Для каждой вершины находим её ребро с минимальным весом (указатель на родителя в дереве)
2. **Нахождение супервершины**
   - У каждого дерева есть корень --- супервершина
   - Супервершина --- одна из пары вершин ссылающихся друг на друга
   - Для каждой вершины протянем какой супервершине она принадлежит



Борувка Pregel: 4 шага

3. **Обновление рёбер**
   - Обновляем рёбра в соответствие с супервершинами соседей
   - Было `e = (v, l, u)`,
     стало `e = (v, l, SuperVertex(u))`
4. **Сжатие до супервершины**
   - Каждая вершина отправляет свои рёбра супервершине
   - Супер вершина сохраняет их, ставя в приоритет при конфликтах ребро с меньшим весом

Алгоритм взят из статьи*.

Optimizing Graph Algorithms on Pregel-like Systems (2014)

Тестовый датасет (дороги)

- SNAP: графы дорог
- 2015: A Generic and Highly Efficient Parallel Variant of Boruvka's Algorithm
- DIMACS: графы дорог США

Тестовый датасет (дороги)

Region	Nodes	Edges
(SNAP)		
California	1,965,206	2,766,607
Pennsylvania	1,088,092	1,541,898
Texas	1,379,917	1,921,660
(DIMACS)		
Western USA	6,262,104	15,248,146
Eastern USA	3,598,623	8,778,114
Great Lakes	2,758,119	6,885,658
California and Nevada	1,890,815	4,657,742
Florida	1,070,376	2,712,798
Colorado	435,666	1,057,066
New York City	264,346	733,846

Описание эксперемента (MS-BFS/Борувка)

Какая реализация Pregel демонстрирует наилучший результат по скорости для «**алгоритма**»?

Насколько отличается ускорение «**алгоритма**» при увеличении числа выделенных ядер для различных реализаций Pregel?

- Реализации Pregel:
  - Pregel+
  - Giraph
  - Spark
- Ход эксперемента:
  - Запустить алгоритм несколько раз
  - Составить информацию о среднем значении и доверительных интервалах
  - Визуализировать результаты

Характеристики вычислительной машины

- **CPU**: `AMD Ryzen 7 4700U`
  - 8 ядер
  - нет hyper-threading
  - все ядра равнозначны
- **RAM**: `16 Gb`
  - без файла подкачки
- **OS**: `Ubuntu MATE 22.04.5 LTS x86_64`