



Санкт-Петербургский государственный университет
Кафедра системного программирования

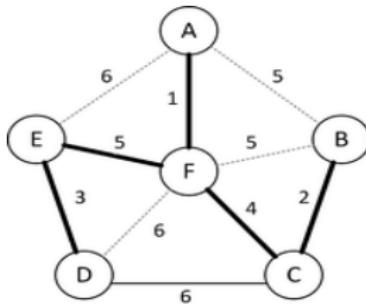
Теория графов, презентация 1

Луконенко Никита Игоревич

Санкт-Петербург
2025

Алгоритм Прима: введение

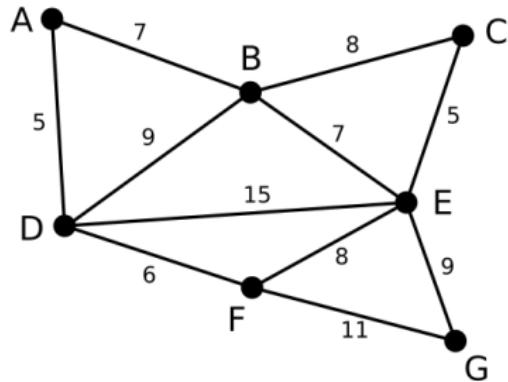
- Остовное дерево — связный ациклический подграф данного связного неориентированного графа, содержащий все его вершины
- Минимальное остовное дерево — основное дерево с минимальным весом



Введение

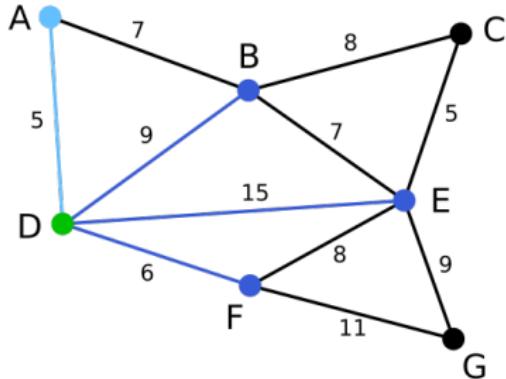
- Есть задачи, которые требуют анализа связей между различными объектами
- Например, при модификации/увеличении сети, есть необходимость анализа её топологии. Минимальное оставное дерево покажет самый дешевый путь, соединяющий все узлы
- Может быть важно для транспортных компаний, или для телекоммуникаций
- Для нахождения минимального оставного дерева в графе существует алгоритм Прима

Пример Прима



Исходный взвешенный граф. Числа возле ребер показывают их веса, которые можно рассматривать как расстояния между вершинами.

Пример Прима



- Выбираем случайную вершину
- Ищем ребро минимальной длины
- Далее есть две вершины, от любой из них ищем ребро минимальной длины до того узла, которого ещё нет в МОД
- Добавляем узел в МОД

Как использовать линейную алгебру?

```
foreach v in V
    do key(v) = INF
    π(v) = nil
weight = 0
choose some arbitrary vertex s in V
key(s) = 0
while mask.nvals() < rows
    eWiseMult(temp, MSTPlus(), d, s)
    u = argmin(temp);
    mask.setElement(u, true);
    idx_weight = d.extractElement(u);
    weight += idx_weight;
    d.update(new_edges)
```

- инструменты линейной алгебры позволяют избавиться от вложенного цикла
- $\pi(v)$ — узел предок
- V — множество вершин
- s — вектор с весами, 0 — если вершина не посещалась и \inf — если посещалась
- d — вектор пар (родитель, вес) с текущими минимальными ребрами

Инструменты

- Существуют различные библиотеки для работы с разреженными графами. В контексте данной работы интересны библиотеки, поддерживающие работу с CPU
- Примерами таких библиотек является GraphBLAS(GBTL) и SPLA, и Suite Sparse (LAGraph), их сравнению и будет посвящена работа
- В GBTL алгоритм Прима есть из коробки, для SPLA и Suite Sparse LAGraph нужно написать реализацию самостоятельно, используя инструменты, предоставляемые библиотекой

Данные

Таблица: Данные¹, используемые в эксперименте

Название графа	Количество узлов	Количество ребер
New York City	264,346	733,846
San Francisco Bay Area	321,270	800,172
Colorado	435,666	1,057,066
Internet	387,240	124,651

- Используются в статье² при сравнении алгоритмов

¹<https://www.diag.uniroma1.it/challenge9/download.shtml>

²<https://userweb.cs.txstate.edu/~mb92/papers/sc23b.pdf>

Алгоритм подсчета треугольников Sandia

Задача подсчета треугольников может решаться в различных предметных областях, где объекты могут формировать группы.
Например в анализе социальных сетей.

Sandia в контексте линейной алгебры

- A — матрица смежности, A^3 — множество путей длины 3 в графе
- Sandia_LL — строится нижнетреугольная матрица (зануляется верхний треугольник).

Тогда формула для подсчёта количества треугольников имеет вид:

$$\sum((U \times U) .* U)$$

- \times — матричное умножение
- $.*$ — поэлементное умножение

Данные для Sandia

Таблица: Данные³, используемые в эксперименте

Название графа	Количество узлов	Количество ребер
graph500-scale19-ef16_adj	335,318	7,729,675
graph500-scale18-ef16_adj	174,147	3,800,348
Amazon0302	262,111	1,234,877
Amazon0312	400,727	2,349,869
Amazon0505	410,236	2,439,437
flickrEdges_adj	105,938	2,316,948

- Используются в статье⁴ при сравнении алгоритмов
- Есть и синтетика, и "натуральные" графы

³<https://www.diag.uniroma1.it/challenge9/download.shtml>

⁴<https://www.osti.gov/servlets/purl/1466485>

Инструменты

- Существуют различные библиотеки для работы с разреженными графами. В контексте данной работы интересны библиотеки, поддерживающие работу с CPU
- Примерами таких библиотек является GraphBLAS(SuiteSparse(LAGraph)) и SPLA, их сравнению и будет посвящена работа
- В SuiteSparse(LAGraph) алгоритм Sandia есть из коробки, для SPLA также есть реализация Sandia

Эксперимент

Цель: сравнение среднего времени работы различных реализаций алгоритмов.

Шаги

- Провести 20 запусков
- Построить доверительные интервалы
- Сравнить результаты
- Вопросы:
 - ▶ Вопрос: на какой из трех библиотек (GBTL, LAGraph или SPLA) окажется быстрее реализация алгоритма Прима?
 - ▶ Вопрос: на какой из двух библиотек (SPLA и Suite Sparse(LAGraph)) окажется быстрее реализация алгоритма Sandia?

Характеристики машины:

- Процессор: Apple Silicon M1 (8 ядер, четыре высокопроизводительных, четыре энергоэффективных)
- ОЗУ: 16 GB
- ОС: macOS Sequoia