# Поиск остовного дерева с заданным числом листьев

Шмаков Александр
группа 22.Б11-мм

# Постановка задачи и актуальность

- **Дано:** Неориентированный связный граф G и число k.
- **Найти:** Остовное дерево T графа G с k листьями.
- **Актуальность:**
  - **Проектирование сетей** (магистрали vs клиенты)
  - **Биоинформатика** (филогенетические деревья)
  - **Схемотехника** (VLSI design)

# Сложность и подходы к решению

- **Сложность:** NP-полная (Garey & Johnson, 1979).
- **Следствие:** Не существует быстрого (полиномиального) точного алгоритма.
- **Обзор подходов:**
  - Аппроксимационные (быстро, но неточно).
  - Параметризованные (FPT) (сложно, эффективно при малом k).
  - Точные экспоненциальные (Backtracking) — наш фокус.

# Выбранные алгоритмы

- **Алгоритм 1: Baseline (Backtracking по рёбрам)**
  - **Принцип:** "Слепой" перебор. Рекурсивно решает: "добавить это ребро или нет?".
  - **Недостаток:** Не использует информацию о цели (k) до самого конца.
- 
- **Алгоритм 2: Improved (Backtracking по ролям вершин)**
  - **Принцип:** Интеллектуальный поиск. Рекурсивно решает: "сделать эту вершину листом или внутренним узлом?".
  - **Ключевые эвристики (в чем "интеллект"):**
    i. **Приоритет важным вершинам:** Первыми рассматриваются вершины с наибольшим числом связей.
    ii. **Раннее отсечение:** Мгновенно отбрасывает ветки, где цель k уже недостижима.
    iii. **Структурная проверка:** Не строит бессвязные части дерева.

# Асимптотическая сложность

- **Baseline: O(2^E * poly(V, E))**
  - Экспонента зависит от числа рёбер E.
- **Improved: O(c^(V-k) * poly(V, E))**
  - Экспонента зависит от числа внутренних вершин V-k.
- **Теоретический прогноз:**
  - Improved должен быть значительно быстрее.
  - Производительность Improved зависит от k, а Baseline — нет.

# Методология эксперимента

- **Эксперимент "Лестница сложности":** Сравнение на графах возрастающей сложности.
- **Гипотеза:** С ростом размера графа разрыв в производительности между Baseline и Improved будет расти экспоненциально.
- **Параметры:**
  - **k:** Для чистоты эксперимента k всегда выбирается как N / 2.
  - **Запуски:** 3 запуска для каждого теста для получения среднего и стандартного отклонения.
  - **Таймаут для Baseline:** 15 секунд.

# Результаты: Эксперимент 1 ('Лестница сложности')

| Название теста | N | k | Время (Baseline, μs) | Время (Improved, μs) | Ускорение |
|---|---|---|---|---|---|
| **Решетка 3x3** | 9 | 4 | 23 ± 5 | 3 ± 1 | 8.6x |
| **Решетка 4x4** | 16 | 8 | 2724 ± 181 | 22 ± 3 | 123.8x |
| **Решетка 5x5** | 25 | 12 | 532690 ± 1724 | 171 ± 7 | 3115.1x |
| **Karate Club** | 34 | 17 | Timeout (>15s) | 38 ± 5 | >> 1000x |

# Анализ результатов эксперимента 1

- **Экспоненциальный разрыв:** Ускорение растет от ~9x на N=9 до ~3100x на N=25.
- **"Стена" для Baseline:** При N=34 наивный подход становится непрактичным.
- **Стабильность Improved:** Низкое стандартное отклонение говорит о предсказуемой производительности.
- **Вывод:** Гипотеза полностью подтверждена. Интеллектуальный поиск на порядки эффективнее.

# Результаты: Эксперимент 2 (Масштабируемость Improved)

| Название теста | N | k | N - k | Время (Improved, μs) |
|---|---|---|---|---|
| **Dolphins** | 62 | 31 | 31 | 965 ± 15 |
| **Dolphins (k → N)** | 62 | 55 | 7 | 275989 ± 1886 |
| **Football** | 115 | 57 | 58 | 762 ± 10 |
| **Football (k → N)** | 115 | 100 | 15 | Timeout (>30s) |

# Анализ результатов эксперимента 2

- **Практическая применимость:** Improved легко решает "средние" задачи ($k \approx N/2$) для графов до 115 вершин менее чем за миллисекунду.
- **Влияние N-k и плотности:**
  - На разреженном графе Dolphins задача с малым N-k=7 решается, но дольше, чем задача со средним N-k=31.
  - На плотном графе Football задача с малым N-k=15 оказалась слишком сложной и ушла в таймаут.
- **Вывод:** Производительность Improved— это сложный баланс между N-k и плотностью графа.

# Итоги

- **Improved на порядки превосходит Baseline.** Это доказано экспериментально.
- **Теория подтверждена практикой.** Результаты согласуются с асимптотическим анализом сложности.
- **Ключ к успеху** — интеллектуальное сокращение пространства поиска.
- **Даже Improved имеет пределы.** Его эффективность зависит не только от N-k, но и от плотности графа, что может сделать даже теоретически "легкую" задачу нерешаемой на практике.

# Source code

Реализацию алгоритмов и код экспериментов можно найти на GitHub: https://github.com/AlexShmak/golang-graphs