

Лабораторная работа № 8 по курсу дискретного анализа: Графы

Выполнил студент группы 08-303 МАИ Арусланов Кирилл.

Условие

Задан неориентированный двудольный граф, состоящий из n вершин и m рёбер. Вершины пронумерованы целыми числами от 1 до n . Необходимо найти максимальное паросочетание в графе алгоритмом Куна. Для обеспечения однозначности ответа списки смежности графа следует предварительно отсортировать. Граф не содержит петель и кратных рёбер.

Формат ввода. В первой строке заданы $1 \leq n \leq 110000$ и $1 \leq m \leq 40000$. В следующих m строках записаны рёбра — пары чисел, номера вершин, соединённых ребром.

Формат вывода. В первой строке вывести число рёбер в найденном паросочетании. В следующих строках — сами рёбра, по одному в строке. Каждое ребро представляется парой номеров вершин; в паре и в строках числа должны быть отсортированы (по возрастанию), а строки — упорядочены по минимальному числу в паре.

Метод решения

Решение основано на классическом алгоритме Куна для двудольных графов (поиск увеличивающих путей с помощью DFS и инвертирование вдоль найденного пути). Основная идея:

1. Выполнить двузначную раскраску графа методом BFS, получив левую и правую доли L и R .
2. Для каждой вершины v из левой доли запускать DFS (функция `tryKuhn`), которая пытается найти путь до свободной правой вершины по чередующимся рёбрам (рёбра вне `matching`, затем — в `matching` и т.д.). Для предотвращения заикливания используется массив посещённых вершин `used`, который очищается перед каждым внешним запуском DFS.
3. Если найден увеличивающий путь, инвертировать соответствующие рёбра (обновить массив `matchR`), что увеличивает размер `matching` на единицу.
4. По завершении получить список пар (левый, правый) — результат.

Для детерминированного результата списки смежности предварительно сортируются (требование условия). Алгоритм гарантированно корректен для двудольных графов.

Описание программы

Программа состоит из одного файла. Основные используемые структуры данных:

- `adj` — вектор списков смежности (1-based номера вершин);
- `color` — вектор для раскраски вершин (0/1);
- `lefts` — список вершин левой доли L ;
- `matchR` — массив соответствий правых вершин в левую долю (значение -1 , если правая вершина свободна);
- `used` — временный массив посещённых вершин (для одного запуска DFS).

Функция `tryKuhn(v)` реализует DFS для поиска увеличивающего пути, возвращая `true` при успешном нахождении и обновлении `matchR`. После основного прохода собираются пары (левый, правый), упорядочиваются и выводятся в требуемом формате.

Дневник отладки

Проблем при разработке не возникало: программа собрана и протестирована локально, прошла чеккер с первого запуска.

Тест производительности

Проведён бенчмарк алгоритма на полном двудольном графе K_{n_L, n_R} с $n_L = n_R$. В таблице ниже включены результаты измерений времени выполнения (в миллисекундах) и нормированная метрика времени относительно теоретического выражения $n_L \cdot E$ (где E — число рёбер). Нормировка даёт оценку константы при асимптотике $T = O(n_L \cdot E)$.

n_L	V	E	matching	time_ms	$\frac{\text{time}}{n_L \cdot E}$ (ms)
50	100	2500	50	0.561	$4.4888 \cdot 10^{-6}$
100	200	10000	100	3.887	$3.8871 \cdot 10^{-6}$
200	400	40000	200	33.226	$4.1532 \cdot 10^{-6}$
400	800	160000	400	189.105	$2.9548 \cdot 10^{-6}$
500	1000	250000	500	341.880	$2.7350 \cdot 10^{-6}$

Анализ. Для полного двудольного графа $E = n_L \cdot n_R = n_L^2$ (при $n_L = n_R$), и теоретическая оценка алгоритма Куна даёт $T = O(n_L \cdot E) = O(n_L \cdot n_L^2) = O(n_L^3)$. Из представленных измерений видно, что время действительно растёт примерно как n_L^3 : при удвоении n_L время возрастает примерно в 8 раз (в пределах погрешностей измерений и влияния аппаратных особенностей). Нормированная величина $\frac{\text{time_ms}}{n_L \cdot E}$ остаётся в одном порядке на интервале значений, что подтверждает согласие практики и теории.

Недочёты

Текущая реализация корректна и работоспособна, однако возможны улучшения по производительности:

- можно реализовать эвристику предварительного жадного сопоставления (быстрая инициализация `matchR` перед основными запусками);
- можно экспериментировать с порядком обхода соседей и порядком стартовых вершин.

Выводы

Реализованный алгоритм Куна надёжно находит максимальное паросочетание в двудольных графах и прост в реализации. В процессе выполнения лабораторной были подтверждены как корректность реализации, так и практическая сложность, совпадающая с теоретической оценкой $T = O(n_L \cdot E)$.