МФТИ, ФПМИ

Алгоритмы и структуры данных, 2-й семестр, весна 2022

Семинар №6–7. Простейшие алгоритмы на графах (2)

Всюду в этом листке (если не оговорено иное) n означает количество вершин в графе, а m — количество рёбер.

1. Докажите, что в связном графе нет мостов, если и только если его можно сильно ориентировать (то есть так ориентировать все рёбра, что по-прежнему из любой вершины можно будет попасть в каждую). Как находить такую ориентацию?

2. Пусть G — неориентированный граф. Требуется каким-нибудь образом ориентировать все рёбра графа. Пусть c_v — число вершин, достижимых из v при данной ориентации. Найдите ориентацию всех рёбер, максимизирующую $\min\limits_{v \in V(G)} c_v$. Асимптотика: $O(n+m)$.

3. Для каждой вершины v графа G за суммарное время $O(n+m)$ определите число компонент в графе $G - v$ (то есть после удаления v и всех инцидентных ей рёбер).

4. В стране X — выборы в парламент. Каждый из n избирателей голосует следующим образом: он сообщает, каких двух представителей он хотел бы видеть в парламенте, а каких двух — не хотел бы. Всего представителей (кандидатов) k. Итог выборов устраивает конкретного избирателя, если в парламент вошёл хотел бы один желаемый для него представитель и не вошёл хотя бы один нежелаемый. За $O(n+k)$ определите, можно ли огласить результат выборов, который бы устроил всех избирателей.

5. Пусть G — связный неориентированный граф. Пара вершин называется *ненадёжной*, если существует ребро, удаление которого приводит к исчезновению всех путей между этими вершинами. Найдите число пар ненадёжных вершин за $O(n+m)$.

6. Дан связный неориентированный граф $G = (V, E)$. Множество вершин A в нём назовём *отказоустойчивым*, если после удаления любого ребра графа G из любой вершины $V \setminus A$ найдётся путь хотя бы в одну из вершин A. Найдите минимальное по мощности отказоутойчивое A. Как найти количество таких минимальных A?

7. (Dynamic Connectivity Problem) В изначально пустом графе — n вершин. К нему поступает q запросов двух видов:

i) добавить неориентированное ребро между вершинами u_i и v_i;

ii) сообщить, есть ли путь между вершинами u_i и v_i.

Ответьте на все запросы за $O((n+q)\sqrt{q})$.

8. Раскраска рёбер графа в два цвета называется *почти сбалансированной*, если для каждой вершины $|c_1(v) - c_2(v)| \leqslant 1$, где $c_i(v)$ — число рёбер цвета i, инцидентных вершине v. Найдите почти сбалансированную раскраску рёбер данного графа G за $O(n+m)$.

9. Имеется набор доминошек, каждая из которых состоит из двух половинок. На половинках написаны числа от 0 до 6. Две доминошки можно состыковать, если на соприкасающихся половинках написаны одинаковые числа. Дано n доминошек с известными числами на них. Можно ли выложить все доминошки в один ряд на стол так, чтобы соседние корректно стыковались между собой? Асимптотика: $O(n)$.

10. Задан список из n городов. Название каждого города представляет собой строку, состоящую из маленьких латинских букв. За линейное время от размера входных данных определите, существует ли такая перестановка городов, что первая буква i-го города равна последней букве $(i-1)$-го для всех подходящих i.

1. Рассмотрим дерево `dfs`. Древесные рёбра направим сверху вниз, а обратные — снизу вверх. Можно показать, что в отсутствие мостов такая ориентация будет сильной.
2. Выделите компоненты рёберной двусвязности. Ответ — это размер максимальной такой компоненты. Для этого нужно ориентировать все мосты по направлению к этой компоненте. Останется воспользоваться задачей 2.
3. Каждая вершина может быть помечена точкой сочленения несколько раз. Каждая такая пометка увеличивает число компонент в $G - v$.
4. Заведите по булевой переменной на каждого кандидата и составьте систему условий в виде формулы в 2-КНФ.
5. Пара является ненадёжной, если вершины лежат в разных компонентах рёберной двусвязности. Для решения задачи достаточно найти размеры всех таких компонент.
6. Решите задачу на дереве. Ясно, что тогда в A нужно сложить все листья и только их. Далее сожмите компоненты рёберной двусвязности.
7. Разбейте все запросы на блоки по $\sqrt{q}$ штук. Обрабатывайте запросы по блокам. После обработки первых k блоков можно склеить вершины по компонентам связности. Внутри блока на каждый запрос второго типа можно запускать `dfs`. Время ответа на такой запрос ограничено число рёбер, добавляемых в блоке, то есть числом $\sqrt{q}$.
8. Введите новую вершину s и соедините её со всеми вершинами нечётной степени. В новом графе степени всех вершин чётны, так что его рёбра можно разбить в объединение эйлеровых циклов. Рёбра на каждом цикле можно покрасить в чередующиеся цвета. Это рассуждение позволяет вывести критерий существования искомой раскраски: в каждой компоненты связности должно быть чётное число рёбер и/или положительное число вершин нечётной степени.
9. Введите граф на вершинах $\{0, 1, \ldots, 6\}$. Сопоставьте доминошке ребро. При чём здесь эйлеров путь?
10. Введите граф на буквах $\{a, b, \ldots, z\}$.