ЧАСТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕДЛЬНОЕ УЧРЕЖДЕНИЕ «СЕВЕРО-КАВКАЗСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ»

ЛЕКЦИЯ 3. ТЕМА 3. ОСНОВЫ ТЕОРИИ ГРАФОВ

1. Основные положения теории графов

Теория графов — раздел дискретной математики, изучающий структуры, состоящие из вершин и соединяющих их рёбер. Формально граф G определяется как пара (V, E), где V — непустое конечное множество вершин, а E — множество рёбер, каждое из которых соединяет пару вершин.

Различают следующие типы графов:

- **Неориентированный граф**: рёбра не имеют направления; ребро между вершинами u и v обозначается как неупорядоченная пара $\{u, v\}$.
- **Ориентированный граф (орграф)**: рёбра имеют направление; дуга из u в v записывается как упорядоченная пара (u, v).
- **Взвешенный граф**: каждому ребру (или дуге) сопоставлено числовое значение — вес (длина, стоимость, пропускная способность и т. п.).
- **Мультиграф**: допускаются кратные рёбра между одной и той же парой вершин.
- **Псевдограф**: разрешены петли (рёбра, соединяющие вершину саму с собой).

Степень вершины v в неориентированном графе — количество инцидентных ей рёбер (петля учитывается дважды). В ориентированном графе выделяют полустепень захода $\deg^-(v)$ (число входящих дуг) и полустепень исхода $\deg^+(v)$ (число исходящих дуг).

Базовые понятия:

- **Смежность**: две вершины смежны, если соединены ребром.
- **Инцидентность**: ребро инцидентно вершине, если оно связывает эту вершину с какой-либо другой.
- **Подграф**: граф, вершины и рёбра которого являются подмножествами вершин и рёбер исходного графа.

2. Маршруты и пути в неориентированных и ориентированных графах

Маршрут в неориентированном графе — последовательность вершин и рёбер, в которой каждое ребро соединяет соседние вершины последовательности.

Путь — маршрут без повторяющихся рёбер; **простой путь** — без повторяющихся вершин (кроме, возможно, начальной и конечной в случае цикла).

В ориентированном графе аналогичными понятиями являются **ориентированный маршрут** и **ориентированный путь**, где движение по дугам возможно только в направлении их ориентации.

Длина маршрута (пути) — количество рёбер (дуг) в нём. Расстояние между двумя вершинами — длина кратчайшего пути между ними. Если пути не существует, расстояние считается бесконечным.

Алгоритмы поиска путей

Алгоритм поиска в ширину (BFS) для нахождения кратчайших путей в невзвешенном графе:

1. Инициализировать очередь, поместить в неё стартовую вершину s, пометить её как посещённую.
2. Пока очередь не пуста:
   - Извлечь вершину u из начала очереди.
   - Для каждой смежной вершины v, ещё не посещённой:
     - Пометить v как посещённую, записать родителя $parent[v] = u$.
     - Добавить v в конец очереди.
3. По массиву $parent$ восстановить путь от s до любой достижимой вершины.

Сложность: $O(|V| + |E|)$.

Алгоритм Дейкстры для поиска кратчайших путей во взвешенном графе с неотрицательными весами:

1. Присвоить начальной вершине s расстояние $d[s] = 0$, остальным — $d[v] = \infty$.
2. Использовать приоритетную очередь, упорядоченную по текущему расстоянию.
3. Пока очередь не пуста:
   - Извлечь вершину u с минимальным расстоянием.
   - Для каждого соседа v вершины u:
     - Если $d[u] + w(u, v) < d[v]$, обновить $d[v]$ и установить $parent[v] = u$.
4. По массиву $parent$ восстановить кратчайшие пути.

Сложность: $O((|V| + |E|) \log |V|)$ при использовании двоичной кучи.

3. Связность графов

Граф называется **связным**, если между любой парой вершин существует путь. В неориентированном графе связность проверяется, например, с помощью BFS или DFS: достаточно запустить обход из одной вершины и убедиться, что все вершины посещены.

Для ориентированных графов различают:

- **Сильно связную компоненту**: множество вершин, в котором для любой пары u, v существуют пути $u \to v$ и $v \to u$.
- **Слабо связную компоненту**: связность в соответствующем неориентированном графе (если игнорировать направления дуг).

Алгоритм выделения компонент связности:

1. Для неориентированного графа:
    - Инициализировать массив посещённых вершин как «не посещено».
    - Для каждой вершины v, если она не посещена, запустить BFS/DFS, помечая все достижимые вершины как принадлежащие одной компоненте.
2. Для сильно связных компонент в орграфе применяется алгоритм Тарьяна или Косарайю:
    - Выполнить DFS, фиксируя время входа и выхода.
    - Обработать вершины в порядке убывания времени выхода на транспонированном графе.
    - Каждая такая серия обходов выделяет одну сильно связную компоненту.

Связность — ключевое свойство для анализа надёжности сетей, маршрутизации и устойчивости систем.

4. Эйлеровы графы

Эйлеров цикл — цикл, проходящий по каждому ребру графа ровно один раз. Граф, содержащий эйлеров цикл, называется **эйлеровым**.

Теорема Эйлера для неориентированных графов: связный граф является эйлеровым тогда и только тогда, когда степени всех его вершин чётны.

Полуэйлеров граф содержит эйлеров путь (не обязательно цикл), проходящий по всем рёбрам ровно один раз; условие: ровно две вершины имеют нечётную степень (они будут началом и концом пути).

Для ориентированного графа:

- Эйлеров цикл существует, если граф сильно связен и для каждой вершины $\deg^+(v) = \deg^-(v)$.
- Эйлеров путь существует, если есть ровно одна вершина с $\deg^+ - \deg^- = 1$ (начало) и одна с $\deg^- - \deg^+ = 1$ (конец), а остальные — сбалансированы.

Алгоритм Флёри для построения эйлерова цикла:

1. Проверить выполнение условий эйлеровости.
2. Начать с произвольной вершины (если есть вершины с нечётной степенью — с одной из них).
3. На каждом шаге выбирать ребро, не являющееся мостом (если есть альтернатива), чтобы не разорвать связность оставшейся части графа.
4. Удалять пройденное ребро и переходить к следующей вершине.
5. Продолжать, пока не будут пройдены все рёбра.

Более эффективный подход — алгоритм на основе DFS с удалением рёбер и построением цикла в обратном порядке (алгоритм Хирхольцера).

5. Деревья и взвешенные графы

Дерево — связный ациклический граф. Основные свойства:

- В дереве с n вершинами ровно $n - 1$ рёбер.
- Между любыми двумя вершинами существует единственный простой путь.
- Добавление любого ребра создаёт ровно один цикл.
- Удаление любого ребра нарушает связность.

Остовное дерево связного графа — подграф, содержащий все вершины исходного графа и являющийся деревом.

Минимальное остовное дерево (MST) во взвешенном графе — остовное дерево с минимальной суммой весов рёбер.

Алгоритм Крускала для построения MST:

1. Отсортировать все рёбра по неубыванию веса.
2. Инициализировать пустое множество рёбер MST.
3. Просматривать рёбра в порядке возрастания веса:
    - Если добавление ребра не создаёт цикла (проверяется с помощью системы непересекающихся множеств — DSU), включить его в MST.
4. Остановиться, когда в MST будет $n - 1$ ребро.

Сложность: $O(|E| \log |E|)$.

Алгоритм Прима для построения MST:

1. Выбрать произвольную стартовую вершину.
2. Поддерживать множество уже включённых в дерево вершин и приоритетную очередь рёбер, соединяющих это множество с остальными.
3. На каждом шаге извлекать ребро минимального веса, ведущее к новой вершине, и добавлять эту вершину в дерево.
4. Повторять, пока все вершины не будут включены.

Сложность: $O(|E| + |V| \log |V|)$ с использованием кучи Фибоначчи; для двоичной кучи — $O((|V| + |E|) \log |V|)$.

Взвешенные графы широко применяются для моделирования транспортных сетей, коммуникационных каналов, распределения ресурсов и других задач оптимизации.