2-sat. Автор опоздал на эту часть лекции, поэтому напишет ее сам позднее.

Мосты и точки сочленения. Назовем мостами такие ребра, при удалении которых граф теряет связность. Аналогичные вершины определим как точки сочленения. Как их найти? Сделаем dfs в неориентированном графе, и построем дерево dfs. Те ребра, по которым мы не переходили, мы назовем обратными. В дереве dfs эти ребра будут идти из вершины в ее какого-то предка.

Что тогда верно про мосты? Из поддерева ребра-моста нет ни одного обратного ребра, которое шло бы выше, чем наше ребро. А найти самое высокое ребро в поддереве можно обычной динамикой. Аналогично с точками сочленения.

Отношения вершинной и реберной двусвязности. Проще всего запомнить так — отношения вершинной двусвязности это отношение на ребрах, а отношение реберной — отношение на вершинах.

Вершины v и u находятся в одной компоненте реберной двусвязности, если существует реберно-простой цикл, содержащий u и v.

Отношение такой двусвязности транзитивно. Можно показать, что если $u \equiv v$ и $v \equiv w$ (при этом v и w соединены ребром), то и $u \equiv w$. Для этого надо склеить два цикла, и найти в них новый — из u в w.

Чтобы найти классы эквивалентности, можно удалить все мосты в графе.

Отношение вершинной двусвязности определяется аналогично, но на ребрах.

Задача о поиске кратчайшего пути. Пусть нам дан граф $G = (V,\ E)$. Каждому ребру задан какой-то вес, а весу пути соответствует суммарный вес всех ребер на пути.

Веса на ребрах могут быть естественными (неотрицательные), или не естественными (разрешаем отрицательный вес ребер). Также отдельный случай для нас — существование циклов отрицательного веса. Кроме того, мы иногда хотим искать любые пути, иногда реберно простые, иногда вершинно простые.

Bfs. Пусть мы хотим найти кратчайший путь, где вес каждого ребра равен 1. Идея такая — мы хотим построить слоистую декомпозицию, где уровню i соответствуют вершины на расстоянии i от стартовой. Алгоритм реализуется на очереди.