Кластеризация

- Под задачей кластеризации обычно понимается ситуация, в которой некую коллекцию объектов требуется разделить на несколько логически связных групп.

Кластеризация – использование графа

- Построение остовных деревьев
- Ó́стовное де́рево графа — это дерево, подграф данного графа, с тем же числом вершин, что и у исходного графа. Неформально говоря, остовное дерево получается из исходного графа удалением максимального числа рёбер, входящих в циклы, но без нарушения связности графа

Кластеризация по компонентам связности

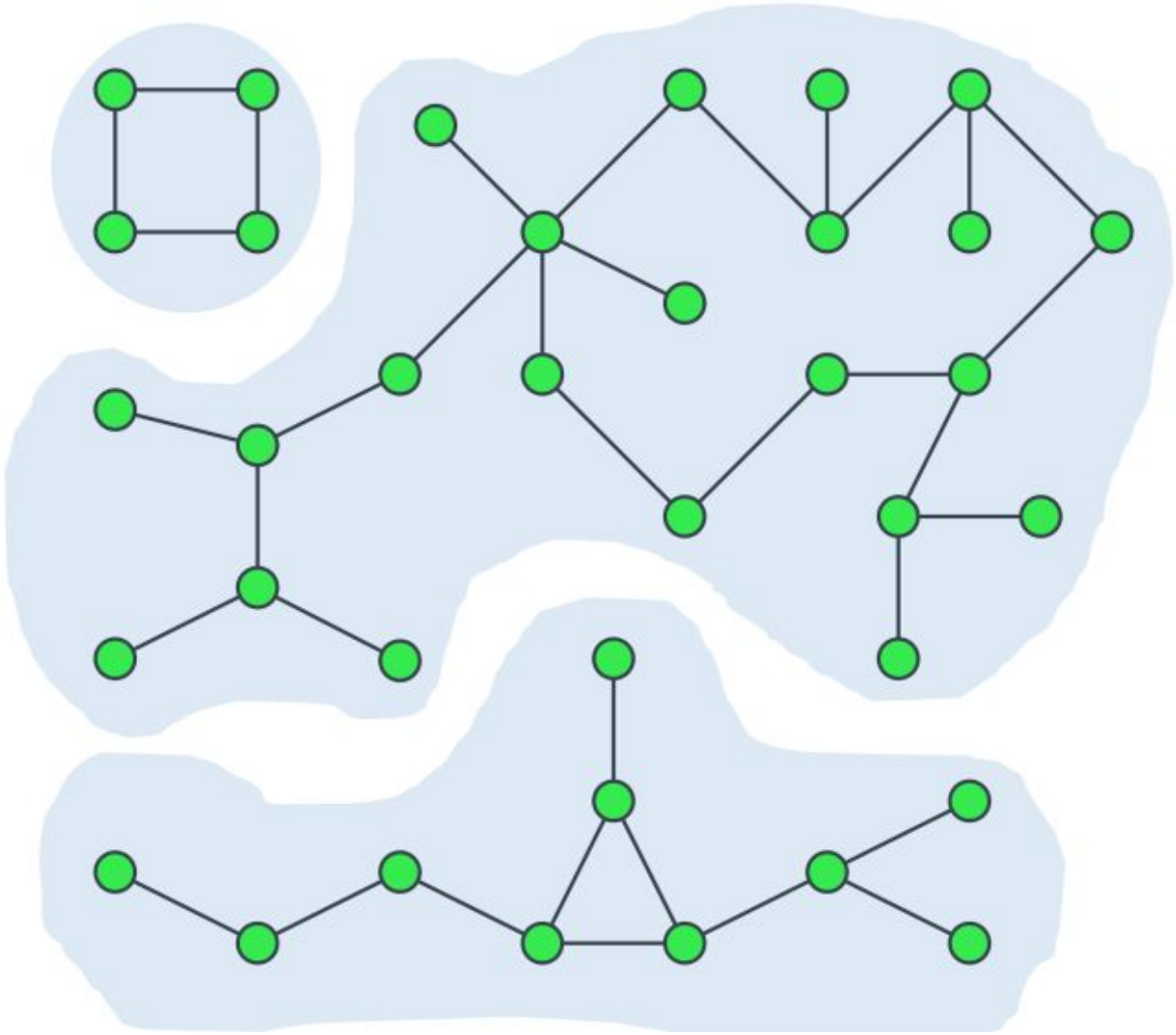
- Граф: вершины соответствуют объектам
- Соединяем ребром объекты, расстояние между которыми меньше
- Выделяем компоненты связности

Кластеризация по максимальному интервалу

- Начнем с создания ребра между ближайшей парой точек; затем создается ребро между парой точек со следующей ближайшей парой и т. д. Далее мы продолжаем добавлять ребра между парами точек в порядке увеличения расстояния d(pi, pj).

- Допустим, имеется множество U из n объектов p1, p2, ..., pn.
- Для каждой пары pi и pj, определяется числовое расстояние d(pi, pj).
- К функции расстояния предъявляются следующие требования: d(pi, pj) = 0;
- d(pi, pj) > 0 для разных pi и pj, а также d(pi, pj) = d(pj, pi) (симметричность).
- Предположим, объекты из U требуется разделить на k групп для заданного параметра k. Термином “k-кластеризация U” обозначается разбиение U на k непустых множеств C1, C2, ..., Ck.

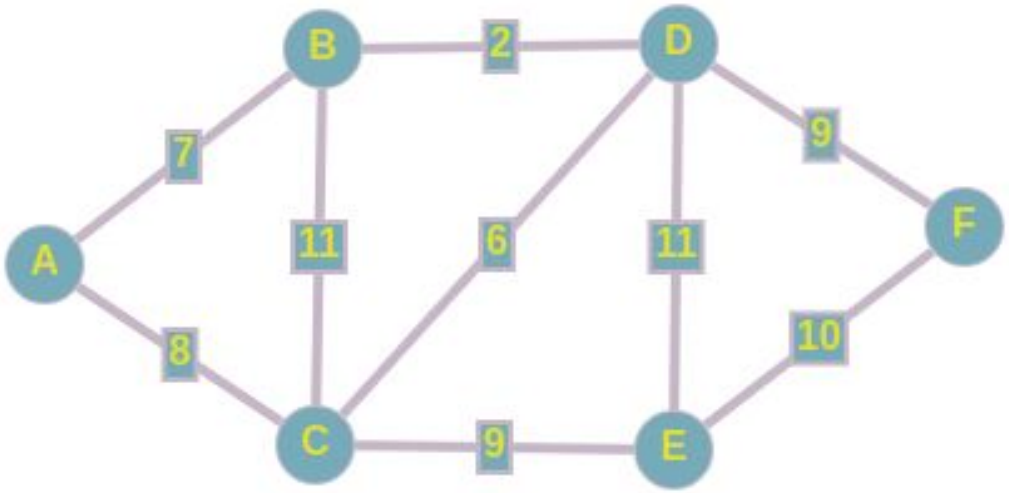
- При таком подходе в процессе расширения графа никогда не образуется цикл, так что H будет объединением деревьев. Добавление ребра, концы которого принадлежат двум разным компонентам, фактически означает слияние двух соответствующих кластеров.

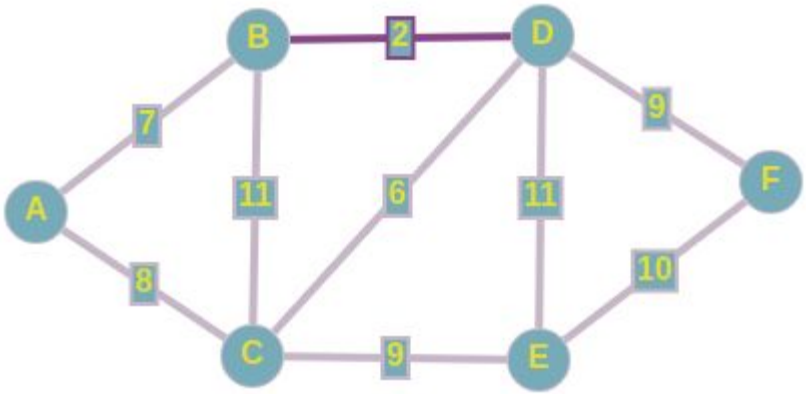


Минимальное остовное дерево

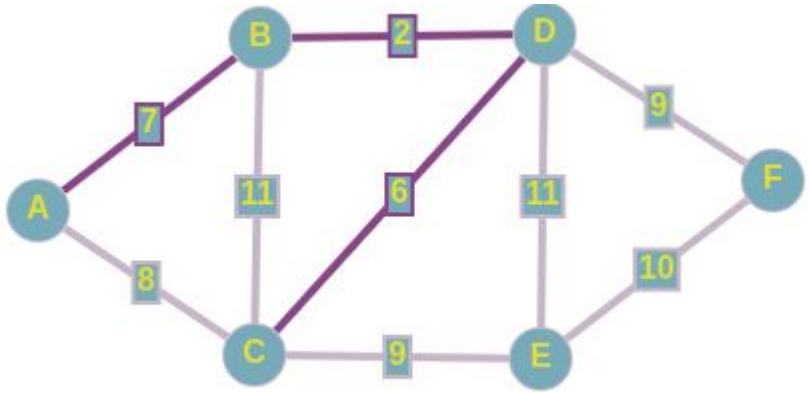
- Ó́стовное де́рево графа — это дерево, подграф данного графа, с тем же числом вершин, что и у исходного графа. Неформально говоря, остовное дерево получается из исходного графа удалением максимального числа рёбер, входящих в циклы, но без нарушения связности графа

- Вначале мы производим сортировку рёбер **по неубыванию** по их весам.
- Добавляем i-ое ребро в наш подграф только в том случае, если данное ребро соединяет две разные компоненты связности, одним из которых является наш подграф. То есть, на каждом шаге добавляется минимальное по весу ребро, один конец которого содержится в нашем подграфе, а другой - еще нет.
- Алгоритм завершит свою работу после того, как множество вершин нашего подграфа совпадет с множеством вершин исходного графа.

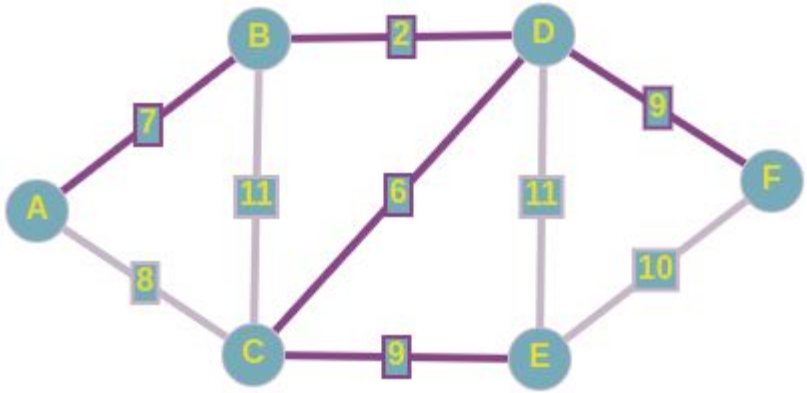




Подграф после добавиления 1-го ребра



Подграф после добавления 2-го и 3-го рёбер



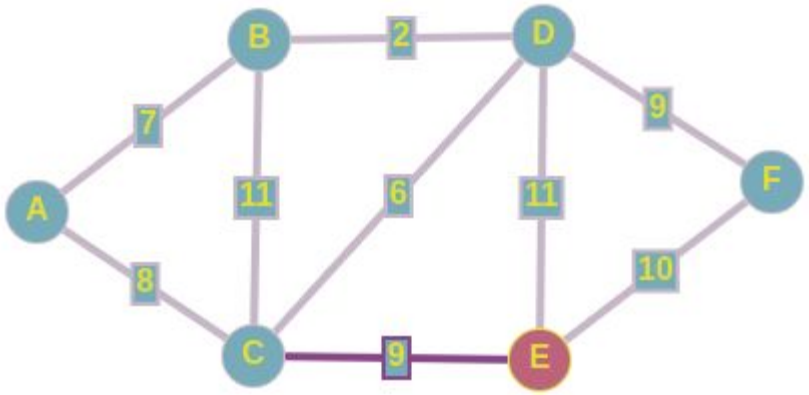
При добавлении в наше остовное дерево ребра A <--> C, как вы можете заметить, образовывается цикл, поэтому мы просто пропускаем данное ребро.
По итогу у нас образовывается следующий подграф, и как вы заметили, мы соединили все вершины ребрами с минимально-возможными весами, а значит, нашли минимальное остовное дерево для нашего исходного графа.

- Вначале, как мы уже раннее говорили, необходимо отсортировать ребра по неубыванию по их весам. Далее каждую вершину можем поместить в свое собственное дерево, то есть, создаем некоторое множество подграфов.
- Дальше итерируемся по всем ребрам в отсортированном порядке принадлежат ли инцидентные вершины текущего ребра разным подграфам,
- если оба конца лежат в разных компонентах, то объединяем два разных подграфа в один

Вариант 2

- На входе так же имеется пустой подграф, который и будем достраивать до потенциального минимального остовного дерева.

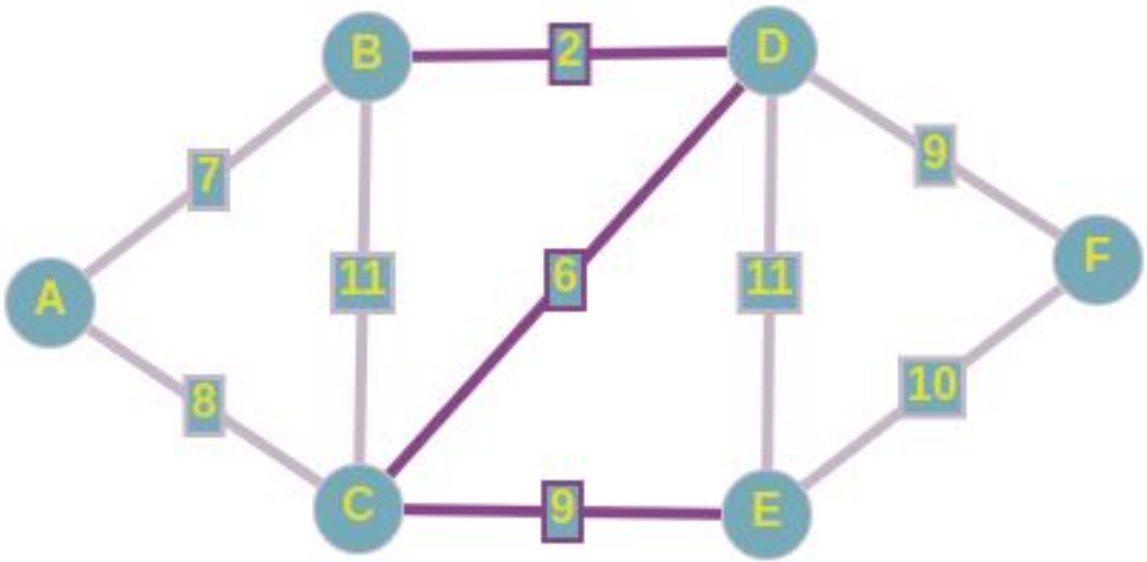
- Изначально наш подграф состоит из одной любой вершины исходного графа.
- Затем из рёбер инцидентных этой вершине, выбирается такое минимальное ребро, которое связала бы две абсолютно разные компоненты связности, одной из которых и является наш подграф. То есть, как только у нас появляется возможность добавить новую вершину в наш подграф, мы тут же включаем ее по минимальмально возможному весу.
- Продолжаем выполнять предыдущий шаг до тех пор, пока не найдем искомое

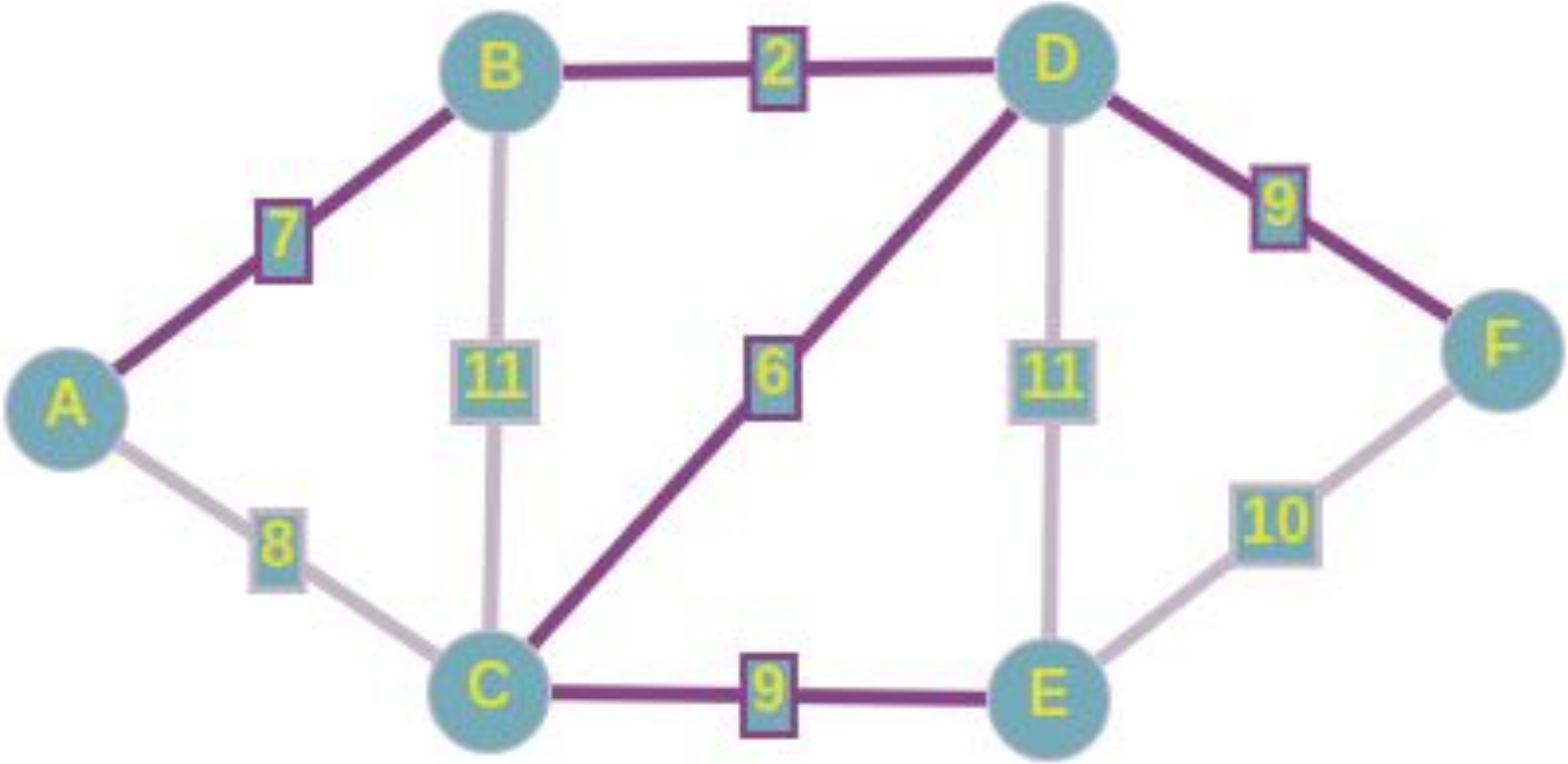


- Выбираем чисто случайно вершину E, далее рассмотрим все ребра исходящие из нее, включаем в наше остовное дерево ребро C <--> E; w = 9, так как данное ребро имеет минимальный вес из всех рёбер инцидентных множеству вершин нашего подграфа.

- Теперь выборка производится из рёбер:
  D <--> C; w = 6
  A <--> C; w = 8
  F <--> E; w = 10
  B <--> C; w = 11
  D <--> E; w = 11

- Добавляем в наш подграф реброD <--> C и по аналогии добаляем ребро D <--> B. Получаем следующее:




B
2
D
7
9
F
A
11
6
11
8
10
C
9
E

- Моделирование сложных сетей, наподобие социальных, или симуляция заболеваний, наподобие коронавируса. Каждый узел может представлять человека или популяцию, а ребра могут представлять вероятность/легкость передачи. В данной модели мы можем попытаться определить или сформировать круговые замкнутые графы.

- Организация и любые иерархические структуры. Графы не обязательно должны представлять циклы и быть циклическими — они также могут выражать иерархию. Например, если вы создадите API для локальной библиотеки, чтобы получать доступ к книгам по их содержанию, то постройте для этого граф. Если вы захотите создать карту сайта, то также используете граф.

- В подразделе МО, именуемом обработкой естественного языка, который занимается моделированием языка, взвешенные представления графов слов и текста чрезвычайно важны, поскольку могут на расстоянии дать понимание, к примеру, слов, принадлежащих к схожему кластеру (“яблоки”, “апельсины”) или означающих схожие вещи.

- Графовые базы данных предназначены для хранения взаимосвязей и навигации в них. Взаимосвязи в графовых базах данных являются объектами высшего порядка, в которых заключается основная ценность этих баз данных. В графовых базах данных используются узлы для хранения сущностей данных и ребра для хранения взаимосвязей между сущностями. Ребро всегда имеет начальный узел, конечный узел, тип и направление.
- Ребра могут описывать взаимосвязи типа «родитель-потомок», действия, права владения и т. п. Ограничения на количество и тип взаимосвязей, которые может иметь узел, отсутствуют.

Выявление мошенничества

- Графовые базы данных позволяют выявлять сложные схемы мошенничества. Анализ взаимосвязей в графовых базах данных дает возможность обрабатывать финансовые операции и операции, связанные с покупками, практически в режиме реального времени. С помощью быстрых запросов к графу можно, например, определить, что потенциальный покупатель использует тот же адрес электронной почты и кредитную карту, которые уже использовались в известном случае мошенничества.
- Графовые базы данных также позволяют без труда обнаруживать определенные шаблоны взаимосвязей, например когда несколько человек связаны с одним персональным адресом электронной почты или когда несколько человек используют один IP-адрес, но проживают по разным физическим адресам.

- **Социальные сети.** Для обнаружения постов мошенников в социальных сетях (для увеличения числа лайков, для перенаправления на вредоносную страницу или на анкетирование) описываются подходы на основе обычных классификаторов, но с учетом графовых признаков: длина распространения «плохого» поста по графу, число лайков и комментов других пользователей по посту, схожесть сообщений, распространяющих пост пользователей, степень узла пользователя, написавшего пост.

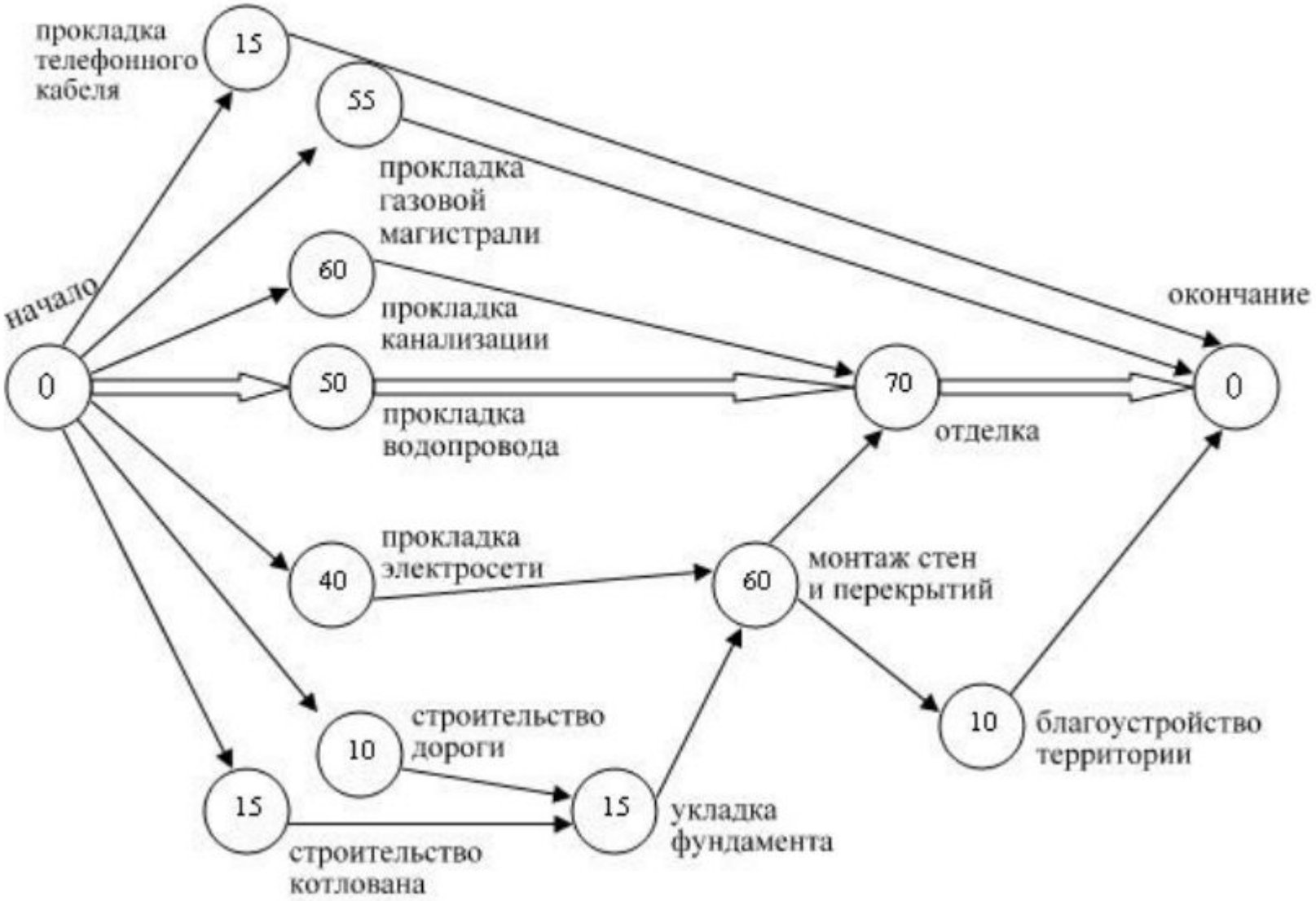
- **Телекоммуникации.** Целью являются люди, пользующиеся услугами бесплатно. Cortes et al. (2002) искали подграфы, тесно связанные с ключевым узлом по параметрам числа и продолжительности звонков. Наблюдения, которые авторы обнаружили: фродовые аккаунты оказались связаны, то есть нарушители или сами звонили друг другу, или звонили на одни и те же телефоны. Второе наблюдение — нарушителей можно обнаружить по схожести их подграфов, определенных предложенным образом.

Сервисы рекомендаций

- Графовые базы данных – хороший выбор для рекомендательных приложений. Используя графовую базу данных, можно хранить в графе взаимосвязи между такими информационными категориями, как интересы покупателя, его друзья и история его покупок. С помощью высокодоступной графовой базы данных можно рекомендовать пользователям товары на основании того, какие товары приобретали другие пользователи, которые интересуются тем же видом спорта и имеют аналогичную историю покупок. Или можно найти людей, у которых есть общий знакомый, но которые еще не знакомы друг с другом, и предложить им подружиться.

- В схемотехнике (топология межсоединений элементов на печатной плате или микросхеме представляет собой граф или гиперграф).
- В химии (для описания структур, путей сложных реакций, правило фаз также может быть интерпретировано как задача теории графов);
- Можно составить граф любой **позиционной игры**: шахмат, шашек, «крестиков – ноликов».

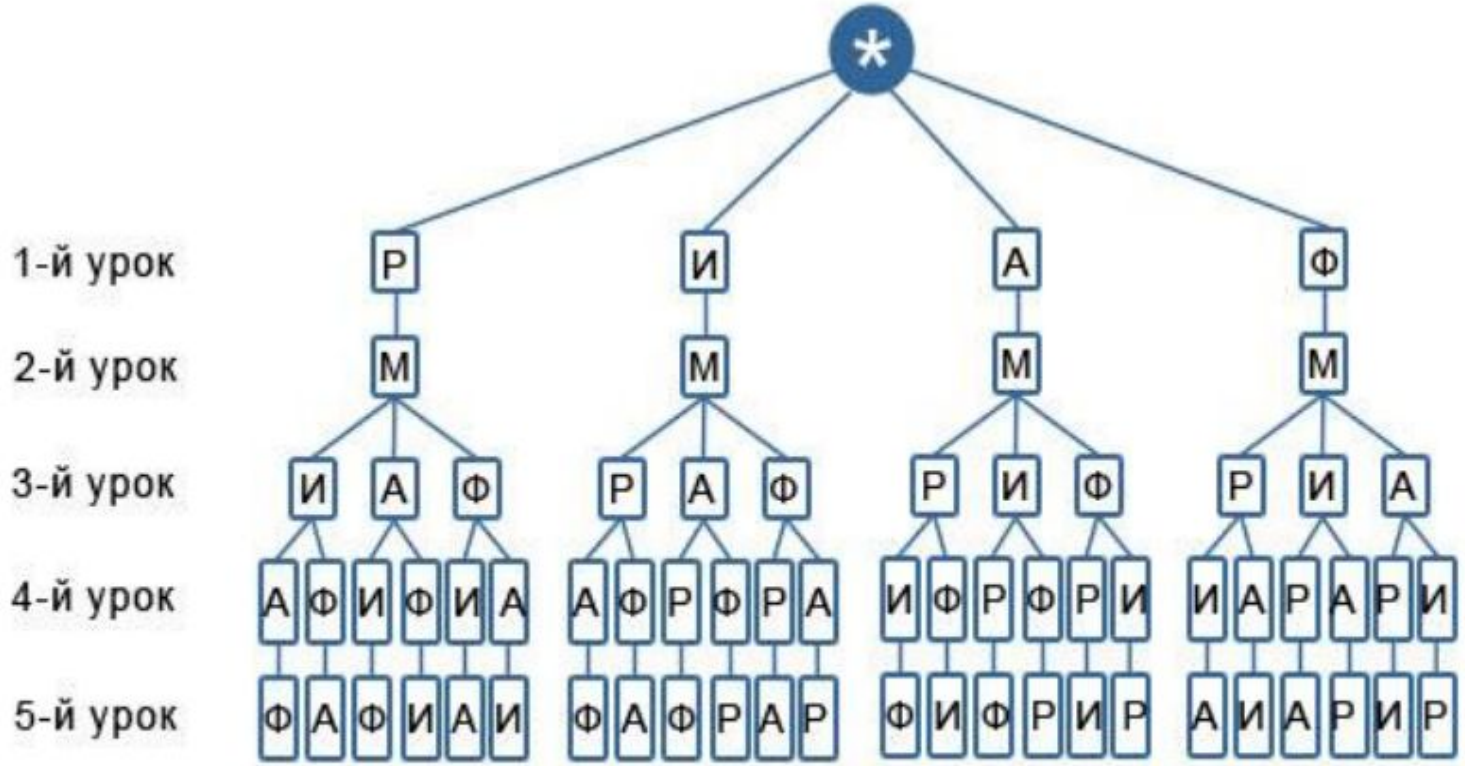
- В строительстве графы используются при планировании проведения работ. Граф, изображенный на чертеже, называется сетевым графиком строительства. В данном случае он составлен для строительства жилого дома. Вершины этого графа обозначают отдельные виды работ на стройке, кроме того, есть еще две вершины: начало строительства и его окончание. Если на ребрах графа нанесены стрелочки, указывающие направление ребер, то такой граф называется направленным.


прокладка телефонного кабеля
15
55
прокладка газовой магистрали
60
прокладка канализации
начало
0
50
прокладка водопровода
70
отделка
0
окончание
40
прокладка электросети
60
монтаж стен и перекрытий
10
строительство дороги
10
благоустройство территории
15
15
укладка фундамента
строительство котлована

- Около вершины графа указаны числа – продолжительность в днях соответствующей работы. Теперь мы можем узнать наименьшую возможную продолжительность строительства. Для этого из всех путей по графу в направлении стрелок нужно выбрать путь, у которого сумма чисел при вершинах наибольшая. Он называется критическим путем (на чертеже большая стрелка). В нашем случае получаем 170 дней. А если сократить время прокладки электросети с 40 до 10 дней, то и время строительства сократится на 30 дней? Нет. В этом случае критический путь станет проходить не через эту вершину, а через вершины, соответствующие строительству котлована, укладке фундамента и т. д. И общее время строительства составит 160 дней, т. е. срок сократится лишь на 10 дней.

Графы и комбинаторика

- Чтобы сделать рациональный выбор, важно не пропустить ни один из них. Для этого надо осуществить перебор всех возможных вариантов или хотя бы подсчитать их количество. Такого рода задачи называют комбинаторными, а соответственно раздел математики – комбинаторика.


1-й урок
2-й урок
3-й урок
4-й урок
5-й урок
*
Р И А Ф
М М М М
И А Ф Р А Ф Р И Ф Р И А
А Ф И Ф И А А Ф Р Ф Р А И Ф Р Ф Р И И А Р А Р И
Ф А Ф И А И Ф А Ф Р А Р Ф И Ф Р И Р А И А Р И Р