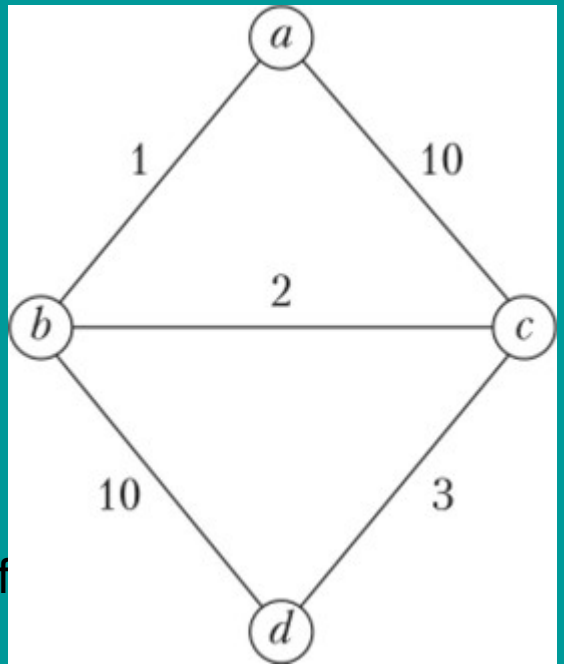
Лекция по дисциплине
«Алгоритмы и структуры данных»

Тема: Графы. **Поиск путей на графах**

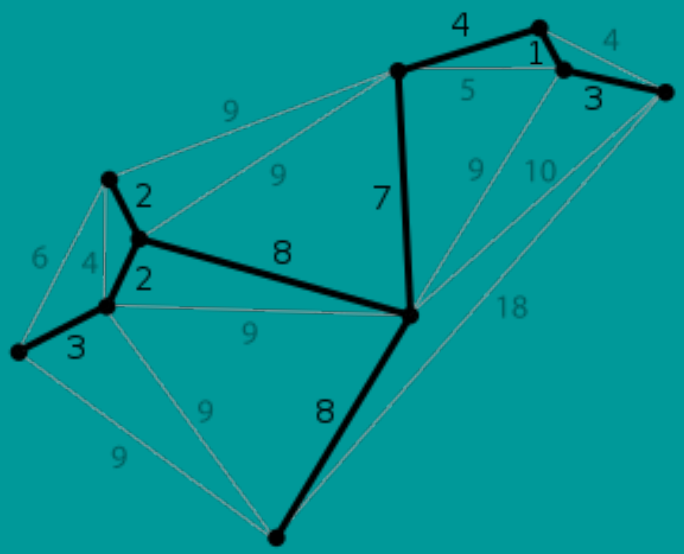
Кратко:
-**Ориентированный граф** (кратко **орграф**) — (мульти) граф, рёбрам которого присвоено
-направление. Направленные рёбра именуются также *дугами*, а в некоторых источниках и просто
-рёбрами. Граф, ни одному ребру которого не присвоено направление, называется
-неориентированным графом или **неорграфом**.
-**Взвешенный граф** — это граф, дугам которого поставлены в соответствие веса, так что дуге
-(xi xj) сопоставлено некоторое число с (xjy Xj) = называемое длиной (или весом, или
-стоимостью) дуги // **Расстояние между вершинами** — это длина кратчайшего пути.
-Если граф имеет цикл (не обязательно простой), содержащий все ребра графа по одному разу,
-то такой цикл называется эйлеровым циклом, а граф — эйлеровым графом.

Определение. Дерево кратчайших путей (SPT) для s – это подграф, содержащий s и все вершины, достижимые из s, образующий направленное Поддерево с корнем в s, где каждый путь от вершины s до вершины и является кратчайшим из всех возможных путей.



Еще определения:
Граф называется вырожденным, если у него нет рёбер.
Вершины называются смежными, если существует ребро, их соединяющее.
Пустым называется граф без рёбер. Полным называется граф, в котором каждые две вершины смежные.
Цикл, в котором все вершины, кроме первой и последней, попарно различны, называется простым циклом.
Граф называется связным, если для любых двух вершин существует путь, соединяющий эти вершины.

Минимальное остовное дерево (или **минимальное покрывающее дерево**) в связанном взвешенном неориентированном графе — это остовное дерево этого графа, имеющее минимальный возможный вес, где под весом дерева понимается сумма весов, входящих в него рёбер.



Алгоритм Дейкстры

Алгоритм на графах, изобретённый нидерландским учёным Эдсгером Дейкстрой в 1959 году. Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании и технологиях, например, его используют протоколы маршрутизации OSPF и IS-IS

Для справки: Эдсгер Вибе Дейкстра (г.ж. 1930 - 2002). Родился в Роттердаме.
Лауреат премии Тьюринга 1972 г.
Технический университет Эйндховена.
Один из основоположников структурного программирования

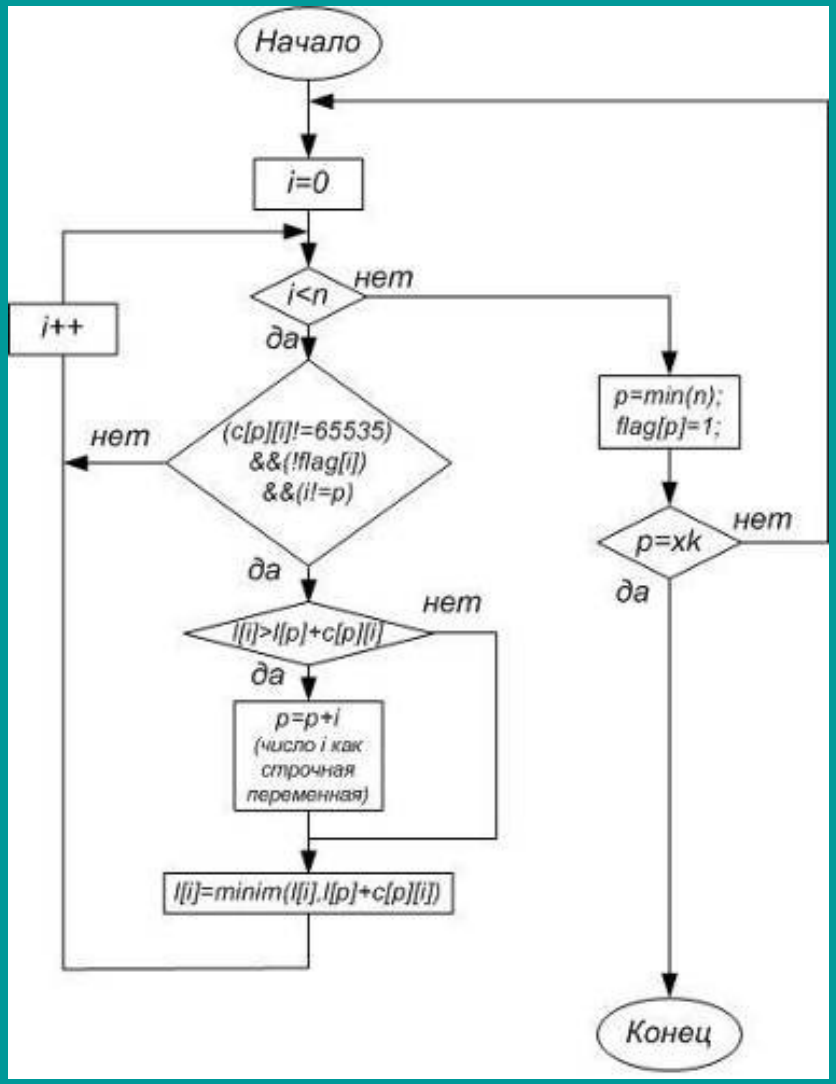
Примеры задач для алгоритма Дейкстры (и не только для него):
-Сеть автомобильных дорог, соединяющая н.п., найти пути от пункта X до всех остальных;
-Авиарейсы между городами, цена перелета из А в Б может быть не равна цене перелета из
-Б в А. Найти маршрут минимальной стоимости (возможно, с пересадками) от аэропорта X до аэропорта Y.

Алгоритм Дейкстры (ищем минимальные расстояния от A до остальных вершин):
1) Каждой вершине ставим метку, равную минимальному известному расстоянию от этой вершины до A.
Задача алгоритма – пошагово уменьшить метки. Если все вершины посещены, то алгоритм завершается.

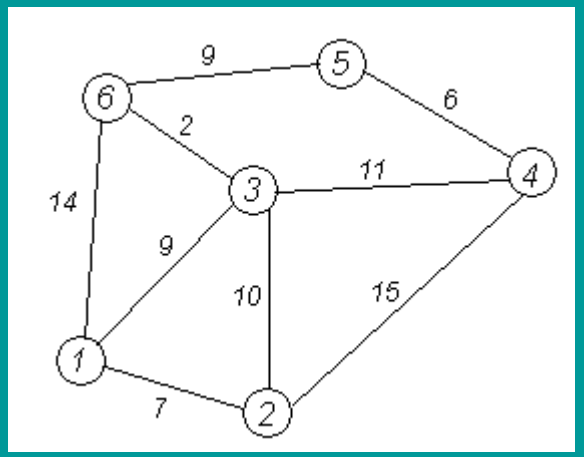
Инициализация:
Метка самой A ставится равной 0, метки остальных равны бесконечности.
Все вершины помечаются как непосещенные.

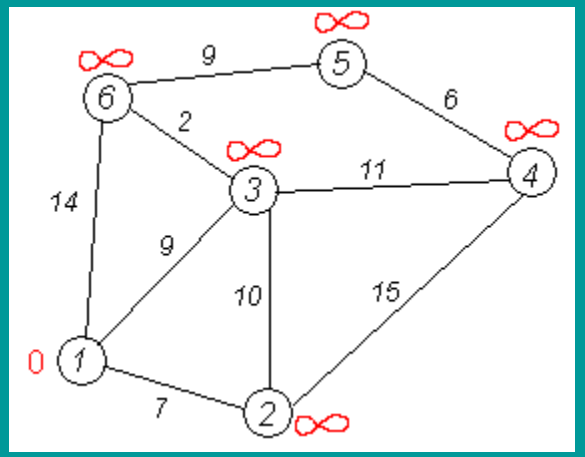
Шаг алгоритма:
Из не посещенных вершин выбирается вершина U, имеющая минимальную метку.
Рассматриваются все возможные маршруты, в которых U является предпоследним пунктом. Вершины в в которые ведут ребра из U, называются соседями этой вершины. Для каждого соседа вершины U, кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки u и длины ребра, соединяющего u с этим соседом.
Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину U как посещённую и
повторяем шаг алгоритма

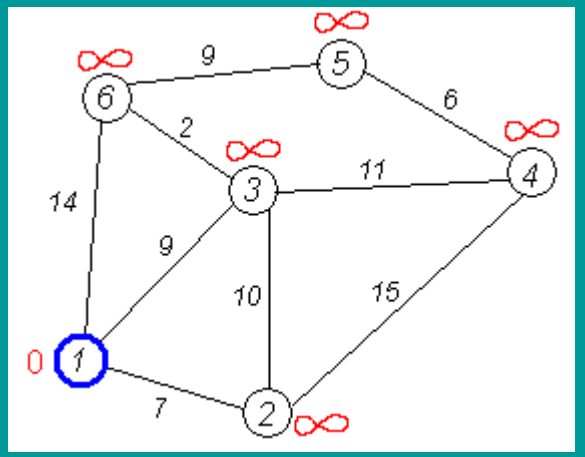
Блок - схема


Начало
i=0
i<n
нет
да
i++
(c[p][i]!=65535)
&&(!flag[i])
&&(i!=p)
нет
да
l[i]>l[p]+c[p][i]
нет
да
p=p+i
(число i как
строчная
переменная)
l[i]=minim(l[i],l[p]+c[p][i])
p=min(n);
flag[p]=1;
p=xk
нет
да
Конец

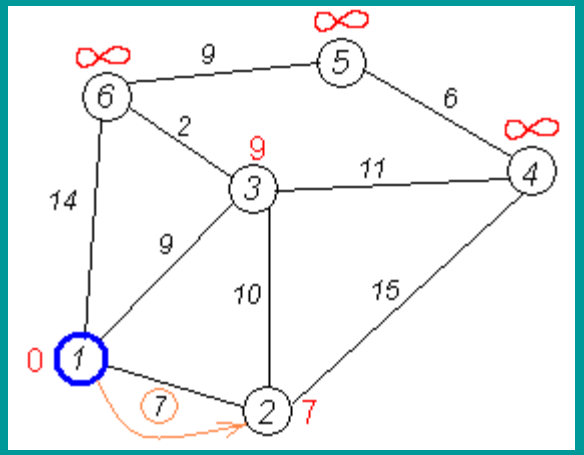
Пример Требуется найти кратчайшие расстояния от вершины (1) до всех остальных.

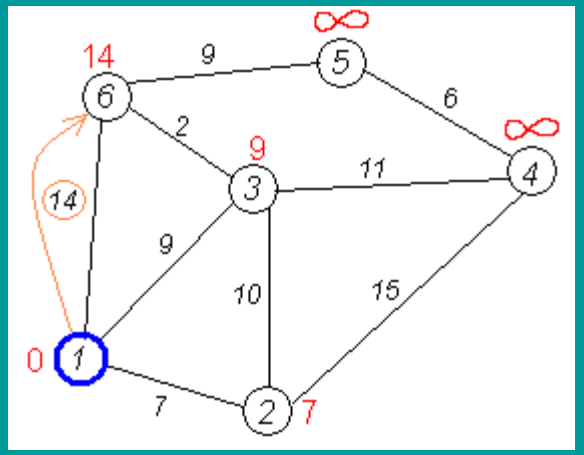


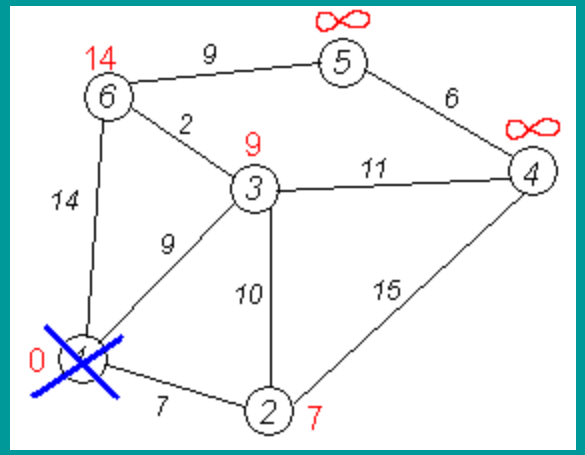




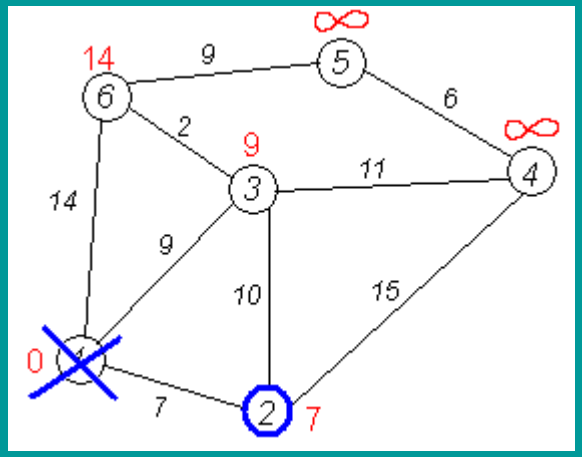
Первый по очереди сосед вершины 1 — вершина 2, потому что длина пути до неё минимальна. Длина пути в неё через вершину 1 равна сумме значения метки вершины 1 и длины ребра, идущего из 1-й в 2-ю, то есть 0 + 7 = 7.Это меньше текущей метки вершины 2, бесконечности, поэтому новая метка 2-й вершины равна 7. Текущее минимальное расстояние до вершины 1 считается окончательным и пересмотру не подлежит. Вычеркнем её из графа, чтобы отметить, что эта вершина посещена.

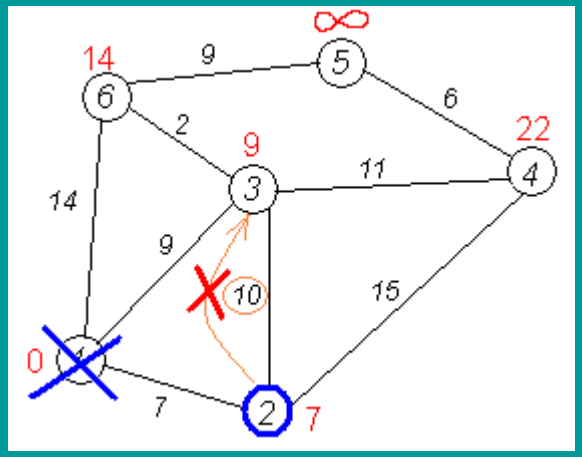


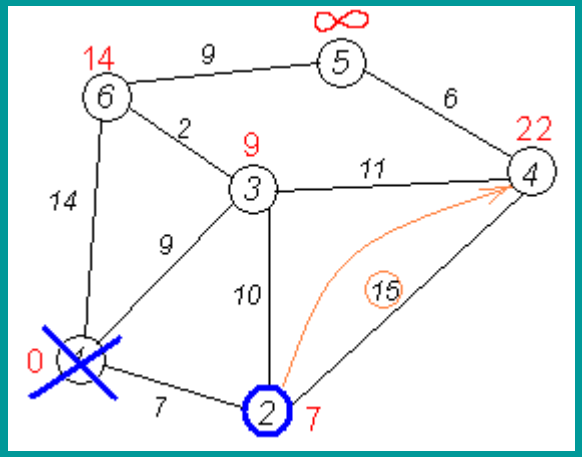




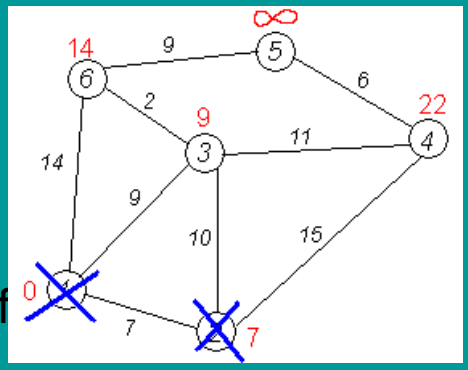
Продолжение **Второй шаг**





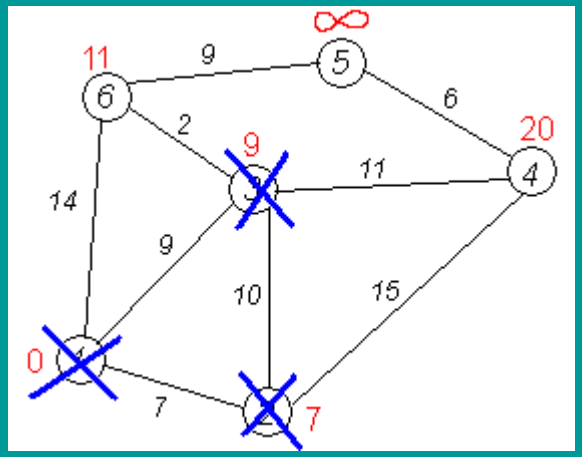


Снова находим «ближайшую» из непосещённых вершин. Это вершина 2 с меткой 7. Снова пытаемся уменьшить метки соседей выбранной вершины, пытаясь пройти в них через 2-ю вершину. Соседями вершины 2 являются вершины 1, 3 и 4. Первый (по порядку) сосед вершины 2 — вершина 1. Но она уже посещена, поэтому с 1-й вершиной ничего не делаем. Следующий сосед — вершина 3, так как имеет минимальную метку. Если идти в неё через 2, то длина такого пути будет равна 17 (7 + 10 = 17). Но текущая метка третьей вершины равна 9, а это меньше 17, поэтому метка не меняется. Ещё один сосед вершины 2 — вершина 4. Если идти в неё через 2-ю, то длина такого пути будет равна сумме кратчайшего расстояния до 2-й вершины и расстояния между вершинами 2 и 4, то есть 22 (7 + 15 = 22). Поскольку 22< бесконечности, устанавливаем метку вершины 4 равной 22.



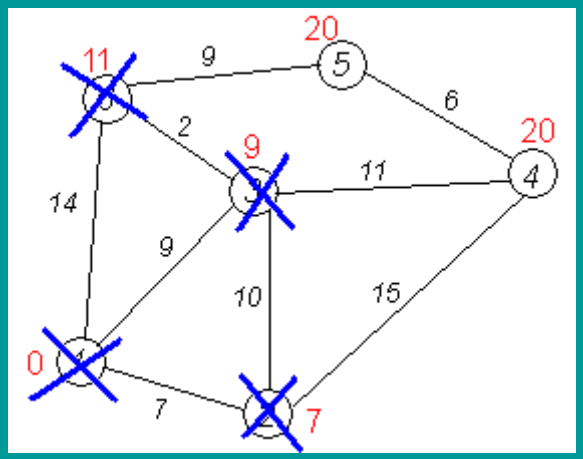
Продолжение **третий шаг**

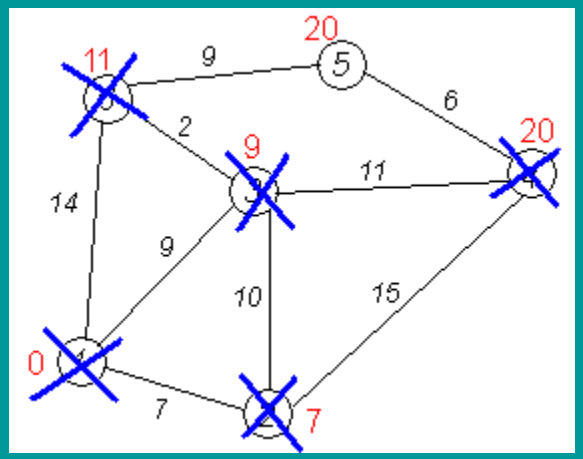
Повторяем шаг алгоритма, выбрав вершину 3. После её «обработки» получим такие результаты:

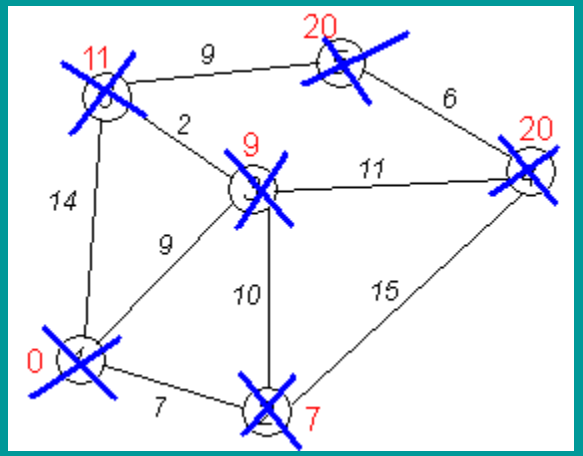


Алгоритм Дейкстры на разных языках программирования
http://rosettacode.org/wiki/Dijkstra'27s_algorithm

И так далее







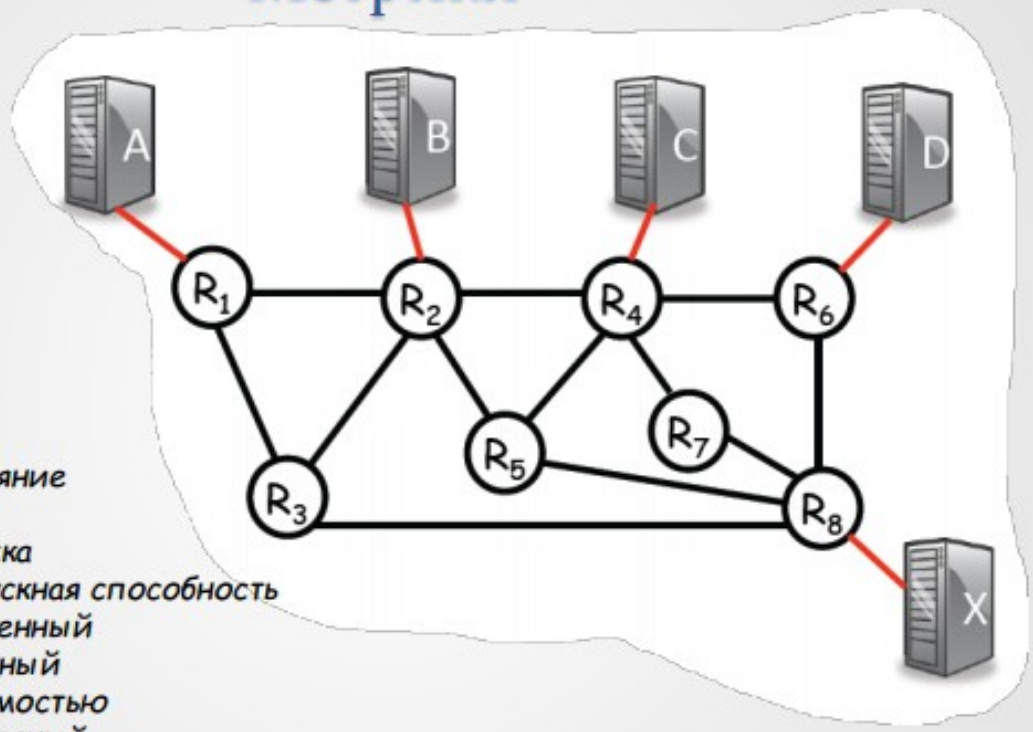
Если в какой-то момент все непосещённые вершины помечены бесконечностью, то это значит, что до этих вершин нельзя добраться (то есть граф несвязный). Тогда алгоритм может быть завершён досрочно.

Алгоритм Беллмана – Форда (алгоритм RIP)

алгоритм поиска кратчайшего пути во взвешенном графе. В отличие от алгоритма Дейкстры, алгоритм Беллмана — Форда допускает рёбра с отрицательным .
был впервые разработан в 1969 году, как основной для сети ARPANET.
(как алгоритм маршрутизации)

```
for v ∈ V
  do d[v] ← +∞
d[s] ← 0
for i ← 1 to |V| − 1
  do for (u, v) ∈ E
    if d[v] > d[u] + w(u, v)
      then d[v] ← d[u] + w(u, v)
return d
```

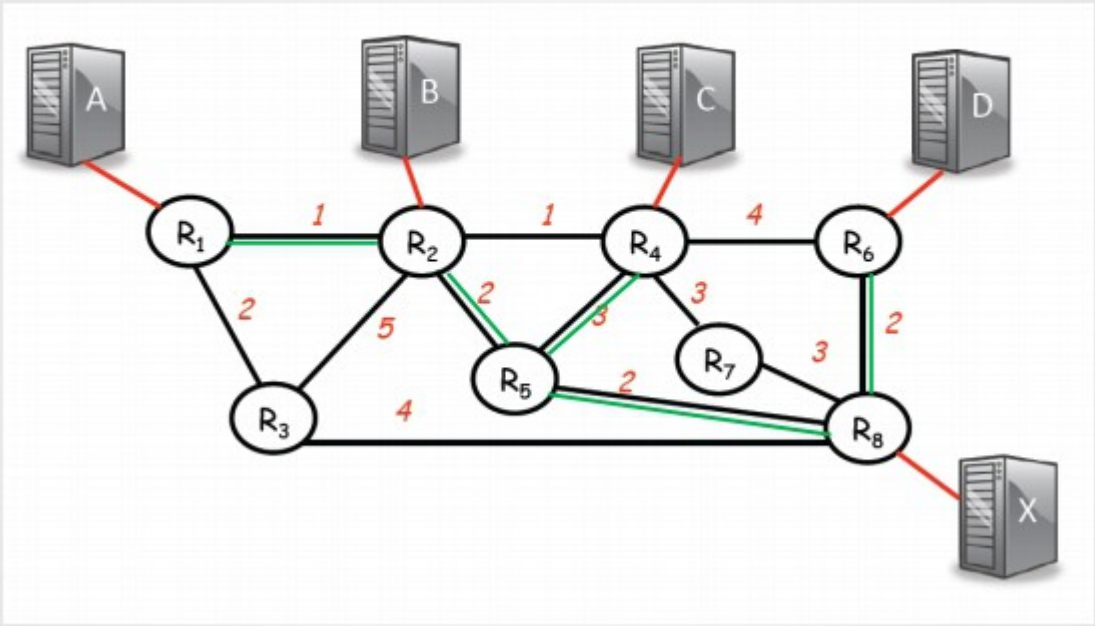
Метрики



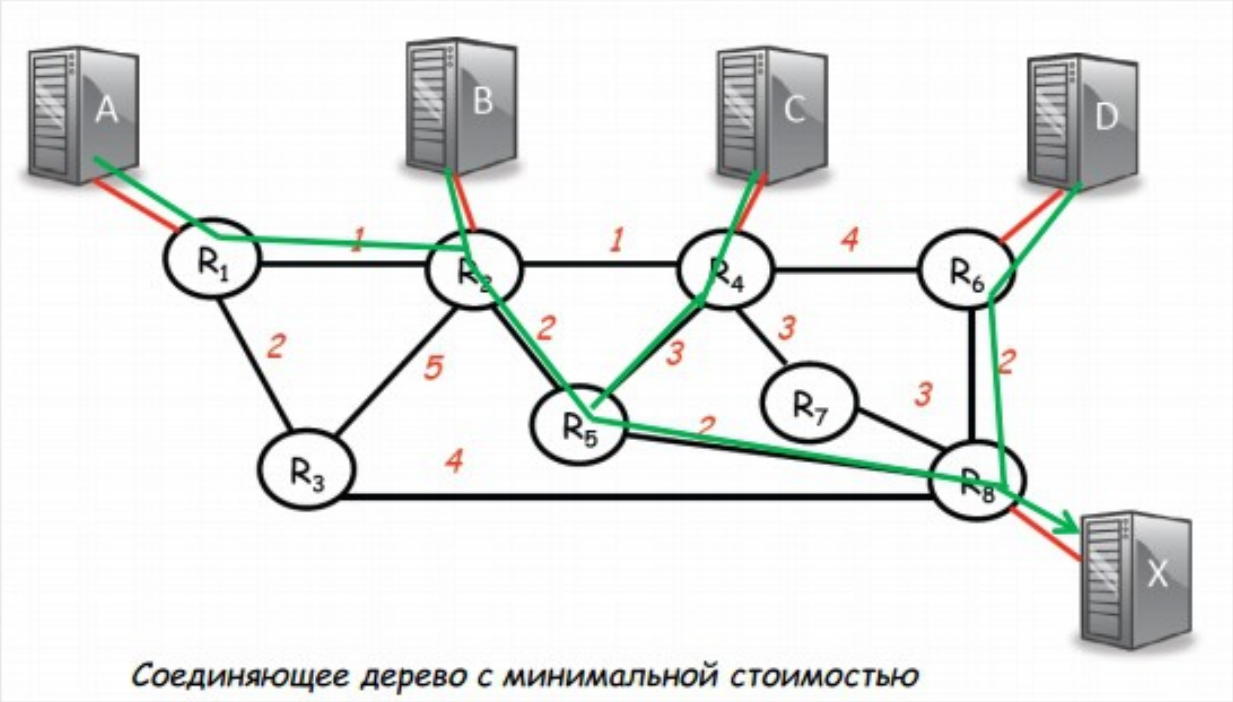
Метрики:

- *мин. расстояние*
- *мин. скачки*
- *мин. задержка*
- *макс. пропускная способность*
- *мин. загруженный*
- *макс. надежный*
- *с мин. стоимостью*
- *макс. безопасный*
- ...

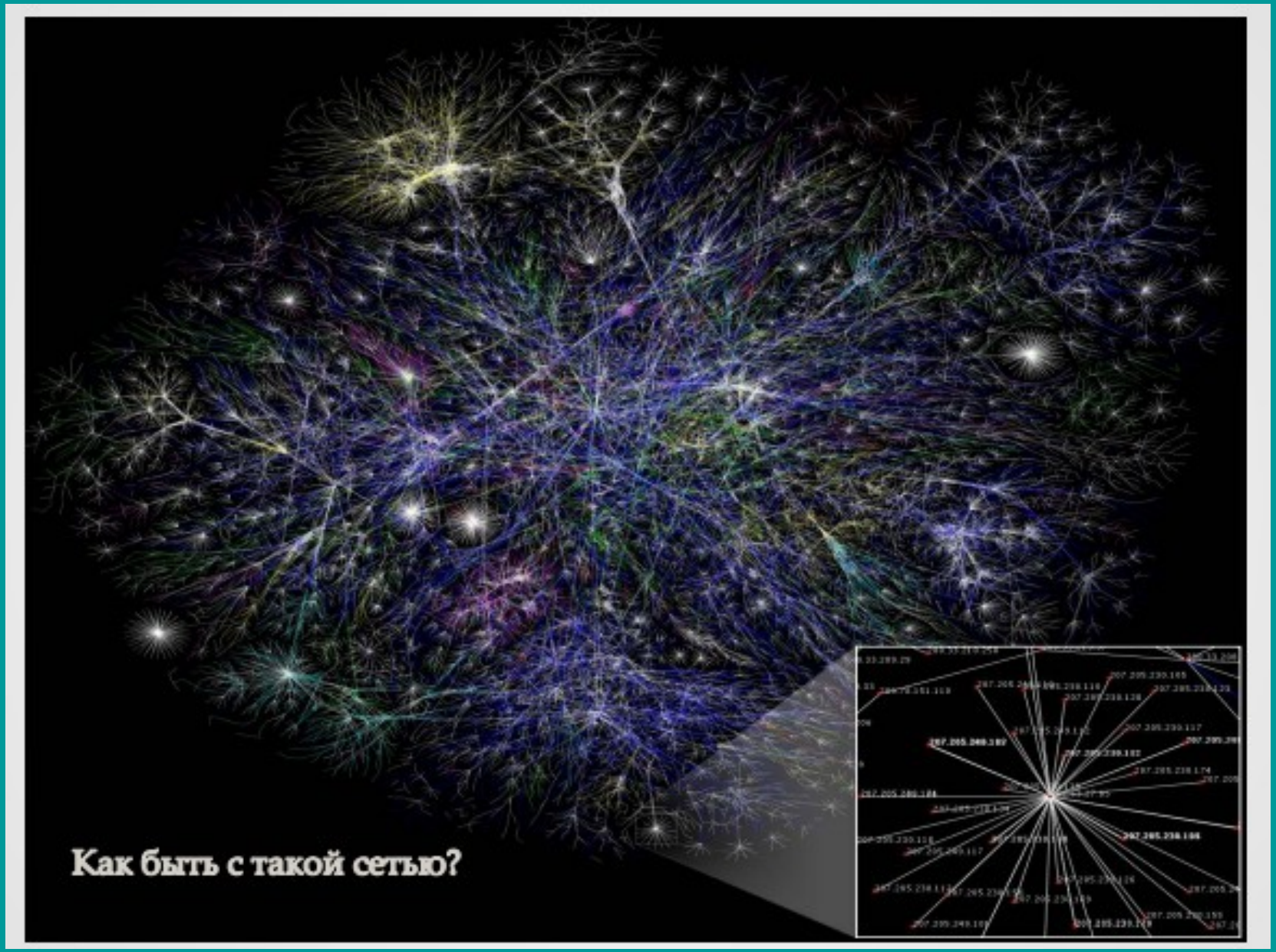
Пример взвешенного графа



Пример взвешенного графа



Соединяющее дерево с минимальной стоимостью


Как быть с такой сетью?

Просто пример компьютерной сети

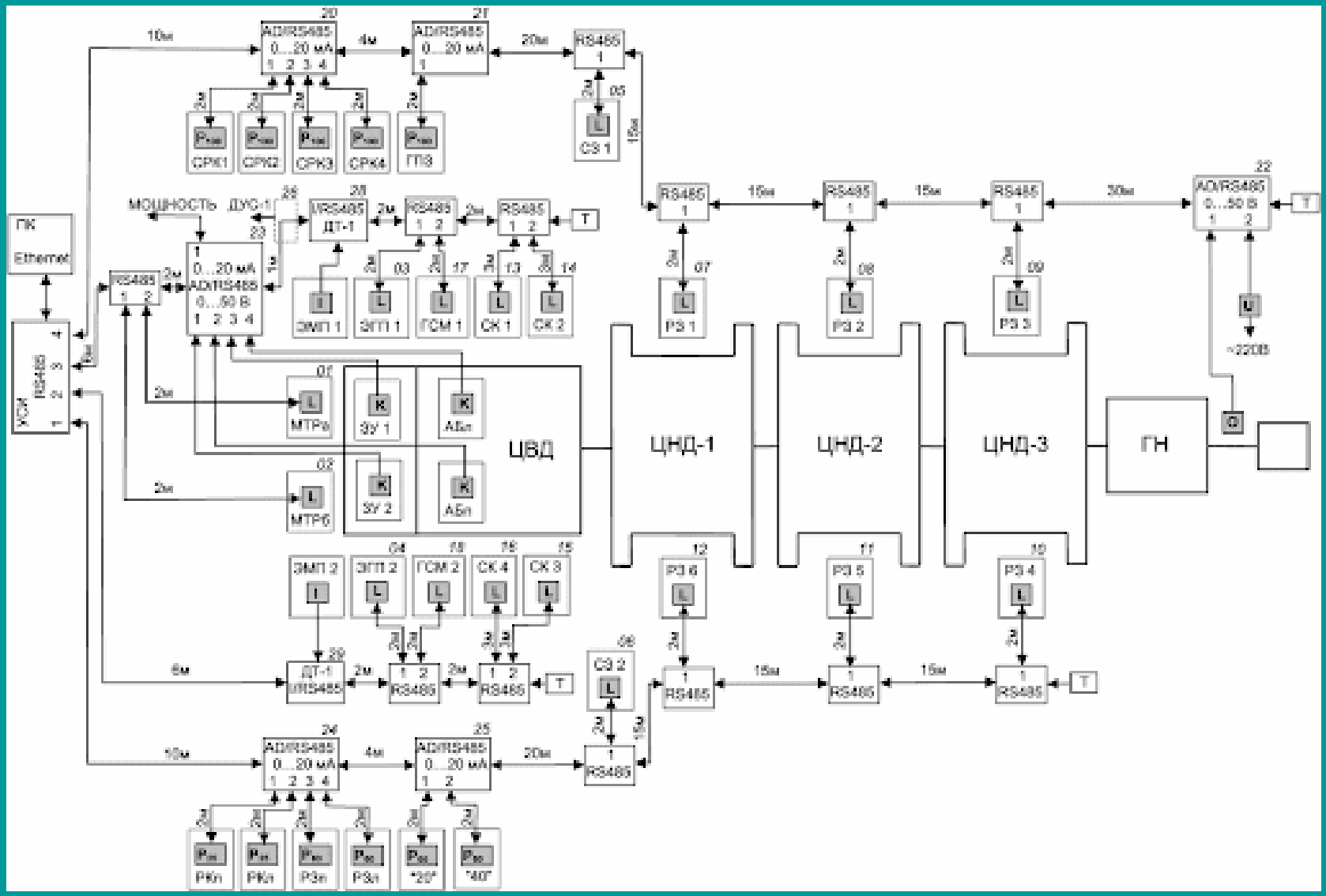


Таблица маршрутизации

Таблица маршрутизации — электронная таблица (файл) или база данных, хранящаяся на маршрутизаторе или сетевом компьютере, описывающая соответствие между адресами назначения и интерфейсами, через которые следует отправить пакет данных до следующего маршрутизатора. Является простейшей формой *правил маршрутизации*.

Таблица маршрутизации обычно содержит:

- **адрес сети или узла назначения**, либо указание, что маршрут является *маршрутом по умолчанию*
- **маску сети назначения** (для IPv4-сетей маска /32 (255.255.255.255) позволяет указать единичный узел сети)
- **шлюз**, обозначающий адрес маршрутизатора в сети, на который необходимо отправить пакет, следующий до указанного адреса назначения
- **интерфейс** (в зависимости от системы это может быть порядковый номер, GUID или символьное имя устройства)
- **метрику** — числовой показатель, задающий предпочтительность маршрута. Чем меньше число, тем более предпочтителен маршрут (интуитивно представляется как расстояние).

https://ppt-online.org/93134

Jump Point Search

Один из самых молодых из алгоритмов поиска путей на графах был представлен в 2011 году. Представляет собой усовершенствованный A*. JPS ускоряет поиск пути, “перепрыгивая” многие места, которые должны быть просмотрены. В отличие от подобных алгоритмов JPS не требует предварительной обработки и дополнительных затрат памяти.

Домашнее задание

https://habr.com/ru/post/331192/ - пример

Алгоритм **A***

(«А-стар» или «А со звездочкой») 1968 г.

Впервые описан в 1968 году Питером Хартом, Нильсом Нильсоном и Бертрамом Рафаэлем. Данный алгоритм является расширением алгоритма **Дейкстры**, ускорение работы достигается за счет эвристики — при рассмотрении каждой отдельной вершины переход делается в ту соседнюю вершину, предположительный путь из которой до искомой вершины самый короткий. При этом существует множество различных методов подсчета длины предполагаемого пути из вершины. Результатом работы также будет кратчайший путь.

Это алгоритм поиска по первому наилучшему совпадению на графе, который находит маршрут с наименьшей стоимостью от одной вершины (начальной) к другой (целевой, конечной)

A* пошагово просматривает все пути, ведущие от начальной вершины в конечную, пока не найдёт минимальный. Как и все информированные алгоритмы поиска, он просматривает сначала те маршруты, которые «кажутся» ведущими к цели. От жадного алгоритма, который тоже является алгоритмом поиска по первому лучшему совпадению, его отличает то, что при выборе вершины он учитывает, помимо прочего, *весь* пройденный до неё путь. Составляющая $g(x)$ — это стоимость пути от начальной вершины, а не от предыдущей, как в жадном алгоритме.

В общем говоря, поиск в глубину и поиск в ширину являются двумя частными случаями алгоритма **A***.

Литература

Дистель Р. Теория графов Пер. с англ. — Новосибирск: Изд-во Ин-та математики, 2002.
Харари Ф. Теория графов. — М.: Мир, 1972.

Интернет-источники:

* https://tproger.ru/articles/pathfindings/