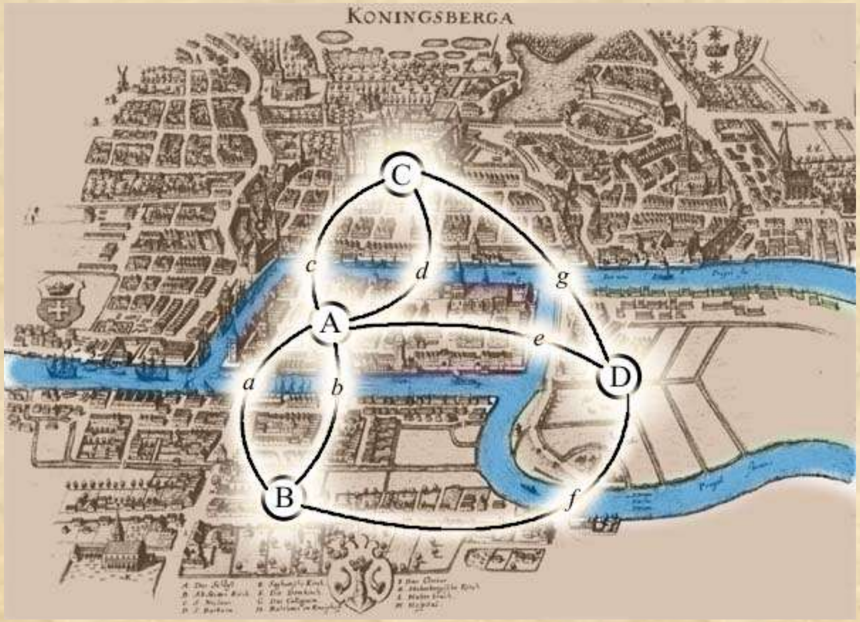
Основы теории графов

Теория графов изучает математические объекты, описывающие связи между элементами конечного множества



ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ГРАФОВ

Графом называется тройка $\langle M,N,T \rangle$, где

M — непустое конечное множество ***вершин***,

N — конечное множество ***дуг***, соединяющих вершины,

T — отображение из *N* в $M \times M$, которое сопоставляет каждой дуге упорядоченную пару вершин. Первая из них называется началом этой дуги, а вторая — концом дуги.

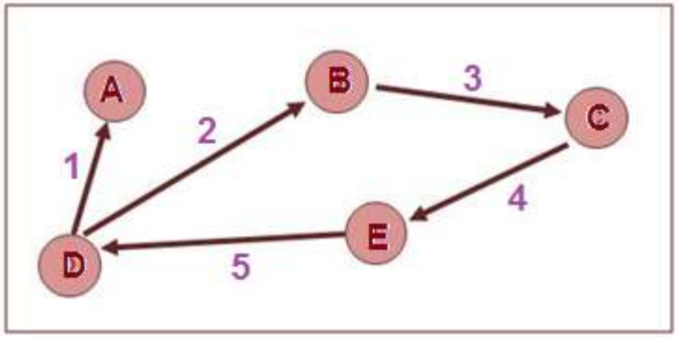
M ={A, B, C, D, E}

N = {1, 2, 3, 4, 5},

T(1) = (D,A); T(2) = (D,B);

T(3) = (B,C); T(4) = (C,E);

T(5) = (E,D);



ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ГРАФОВ

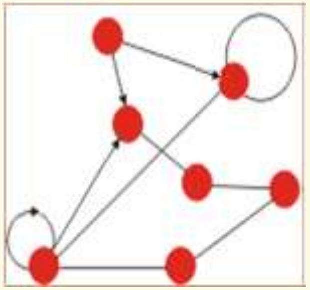
Ребро – неориентированная дуга.

Граф называется неориентированным**,** если каждая его дуга не ориентирована, ориентированным (орграфом), если ориентированы все его дуги, смешанным, если есть как ориентированные, так и неориентированные дуги.

Петля – это дуга, начальная и конечная вершина которой совпадают.

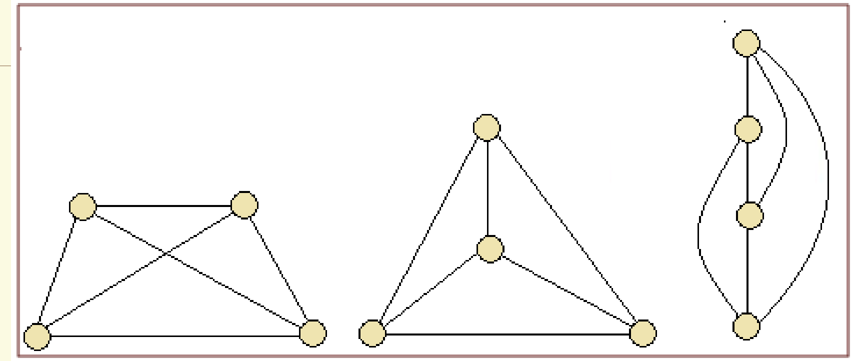
Вершины, прилегающие к одному и тому же ребру, называются смежными.

Полным называется граф, в котором каждые две вершины смежные.



Наглядное представление

Граф можно изобразить графически: сопоставить его вершинам точки, а дугам – линии, идущее от начальной вершины к конечной.



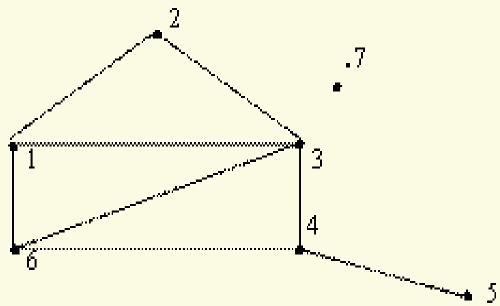
Одному графу можно сопоставить несколько графических представлений.

Изображение призвано лишь показать, какие пары вершин соединены рёбрами, а какие — нет.

ВАЖНО: Форма линии и расположение вершин на рисунке произвольны.

2. СТЕПЕНЬ ВЕРШИНЫ

Степенью вершины называется число ребер графа, которым принадлежит эта вершина. Еще называют его ***валентностью*** и обозначают ***d(v)***, **deg*(v)***. Вершина графа, для которой ***d(v)=0***, ***является изолированной,*** *если* ***d(v)=1***, то ***висячей***.



Deg(6)=3, deg(5)=1, 5 – висячая вершина
N(3)={2,1,6,4}, deg(7)=0, 7 – изолированная вершина

Рис 2.1

Вершина называется нечетной, если *d(v)* – ***нечетное число,*** четной если *d(v)* – ***четное число***. Степень каждой вершины полного графа на единицу меньше числа его вершин.

3. Маршруты, цепи, циклы

Маршрутом в графе называется чередующаяся последовательность вершин и ребер, в которой любые два соседних элемента инцидентны:

$$v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$$

Если $v_0 = v_k$, то *маршрут замкнут,* в противном случае *открыт.*

Если все ребра различны, то маршрут называется *цепью.*

Если все вершины различны, то маршрут называется *простой цепью.* В цепи $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ вершины v_0 *и* v_k называются концами цепи, т. е. цепь концами v_0 *и* v_k соединяет вершины v_0 *и* v_k. Цепь, соединяющая вершины v_0 *и* v_k, обозначается (v_0 *и* v_k). Очевидно, что если есть цепь, соединяющая вершины v_0 *и* v_k - простая цепь, соединяющая эти вершины.

Замкнутая цепь называется циклом, *замкнутая простая – простым циклом,* число циклов обозначается *z(G)*. Граф без циклов – *ациклический.*

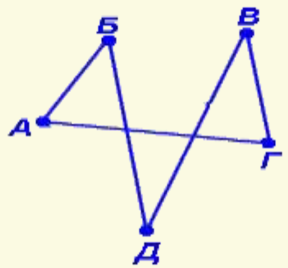
Длинной маршрута называется количество ребер в нем (с повторениями).

Если маршрут $M = v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$, то длина маршрута *M* равна *k*, обозначается $|M| = k$.

4. Связность графов

Две вершины графа *называются связными,* если существует соединяющая их простая цепь. В противном случае две вершины называются *не связными.*

Граф называется связным, если каждые две вершины связные.



Так, на рисунке любая пара вершин, взятая из набора А,Б,В,Г,Д ,будет связной, т.к. от любой из них к любой можно "пройти" по ребрам графа.

Рис 4.1

Граф *называется несвязным,* если хотя бы две его вершины несвязные.

Другая же пара вершин из набора Е,Ж,З, не будут связными, так как от одной к другой "пройти" по ребрам не удается.

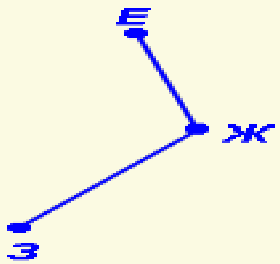
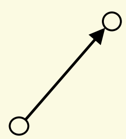


Рис 4.2

5. Ориентированные графы

Если элементы множества ***E*** графа **G(V, *E*)** – упорядоченные пары, то граф называется ***ориентированным*** или ***орграфом.***

Ребро графа называется ориентированным, если одну вершину ребра считают началом ребра а другую концом, на рисунке изображают стрелкой между вершинами. Граф у которого все ребра ориентированы – ***ориентированный.***



Ориентированное ребро

Неориентированное ребро

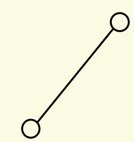


Рис 5.2

Одна и та же вершина ориентированного графа может служить началом для одних ребер и концом для других, поэтому различают две степени вершины:

- *Степенью выхода* вершины орграфа – число ***выходящих*** из вершины ребер;
- *Степенью входа* вершины орграфа – число ***входящих*** в вершину ребер.

В орграфах в зависимости от сочетания степеней входа и выхода для данной вершины рассматриваются три случая:

- ***Изолированной*** вершиной называется вершина у которой степень входа и степень выхода равны 0;
- ***Источником*** называется вершина, степень выхода которой положительна, а степень входа равна 0;
- ***Стоком*** называется вершина, степень входа которой положительна, а степень выхода равна 0.

Путем в ориентированном графе называется последовательность ориентированных ребер.

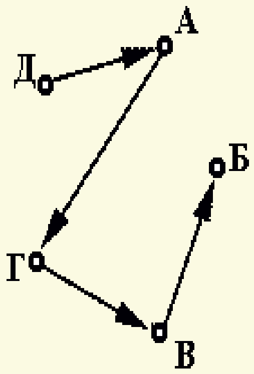


Рис 5.3

Простым путем в ориентированном графе называется путь, в котором ни одна вершина не содержится более одного раза (Рис 5.3). На рис 5.4 изображен не простой путь.

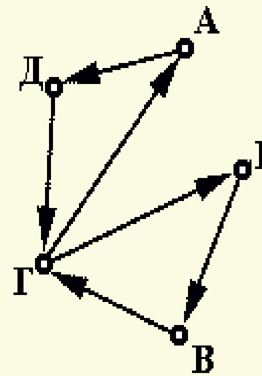
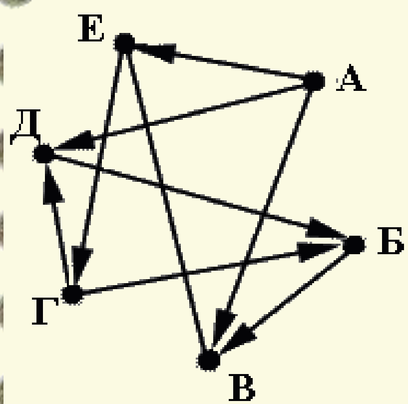


Рис 5.4

Замкнутый путь в ориентированном графе называется ориентированным циклом или контуром. Длиной пути называется число ребер в этом пути.

Полным ориентированным графом называется граф, каждая пара вершин которого соединена в точности одним ориентированным ребром. Если ребра полного графа неориентированные, то граф соответственно будет полным неориентированным.



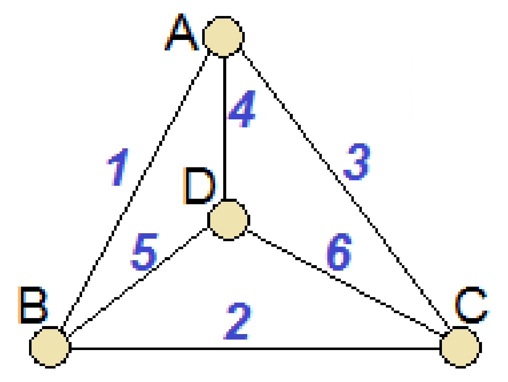
Ориентированный граф

Петлей называется ребро, у которого начальная и конечная вершины совпадают. Петля обычно считается неориентированной.

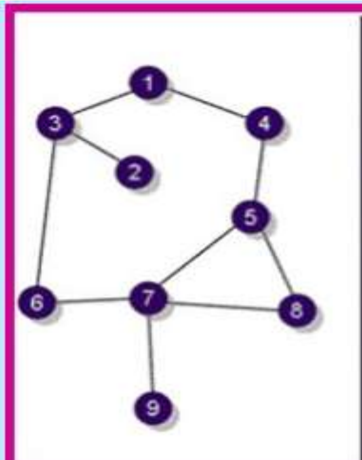
Мультиграфом называется граф, в котором пара вершин соединяется несколькими различными ребрами. Для ориентированного мультиграфа вершины v_i и v_j могут соединятся несколькими ребрами в каждом из направлений.

Табличное представление

	A	B	C	D
1	+	+		
2		+	+	
3	+		+	
4	+			+
5		+		+
6			+	+



	A	B	C	D
A		1	3	4
B	1		2	5
C	3	2		6
D	4	5	6	



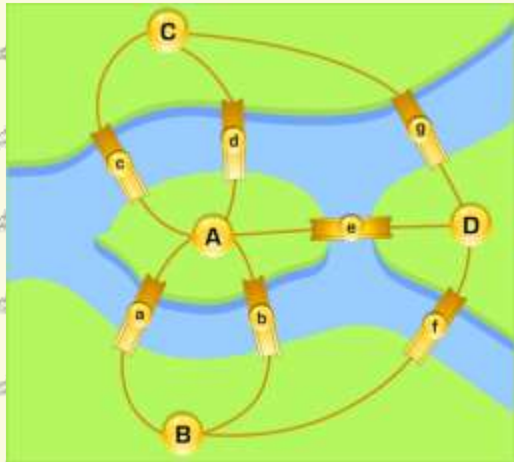
	1	2	3	4	5	6	7	8	9
1			1	1					
2			1						
3	1	1				1			
4	1				1				
5				1			1	1	
6			1				1		
7					1	1		1	1
8					1		1		
9							1		

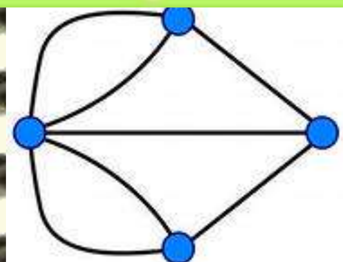
Цикл Эйлера

Долгое время отдельные задачи теории графов появлялись как занимательные задачи, которые легко формулиро-вались, но были почему-то очень трудны для решения.

Среди таких задач наибольшей известностью пользуется задача о семи кёнигсбергских мостах: **в городе Кёнигсберге в начале XVIII века было семь мостов. Возник вопрос: возможна ли такая прогулка, в которой путь пройдет по всем мостам, и по каждому мосту ровно один раз.**

Этот вопрос был предложен знаменитому Леонарду Эйлеру, и в 1735 г. он эту задачу решил: нельзя.

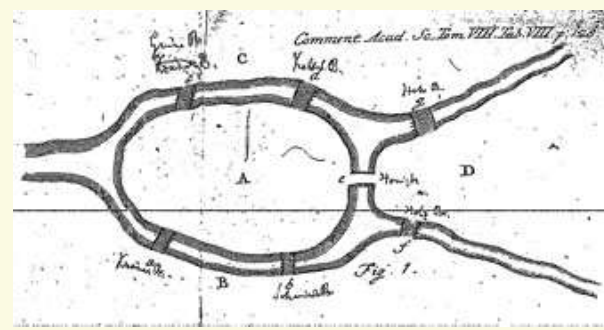




В ходе рассуждений Эйлер пришёл к следующим выводам:

- **Число нечётных вершин** (к которым ведёт нечётное число рёбер) **графа должно быть чётно. Не может существовать граф, который имел бы нечётное число нечётных вершин.**
- **Если все вершины графа чётные, то можно, не отрывая карандаша от бумаги, начертить граф, при этом можно начинать с любой вершины графа и завершить его в той же вершине.**
- **Граф с более чем двумя нечётными вершинами невозможно начертить одним росчерком.**

Граф кёнигсбергских мостов имел четыре нечётные вершины (то есть все), следовательно, невозможно пройти по всем мостам, не проходя ни по одному из них дважды.

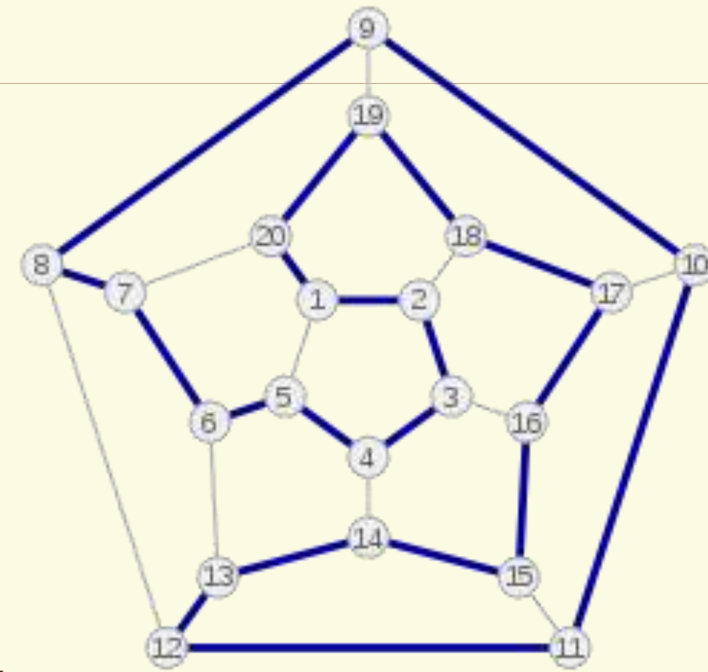


Карта из статьи 28-летнего Эйлера в трудах Санкт-Петербургской Академии наук

Созданная Эйлером теория графов нашла очень широкое применение: например, её используют при изучении транспортных и коммуникационных систем, в частности, для маршрутизации данных в Интернете.

Гамильтонов цикл

Уильям Роуэн Гамильтон



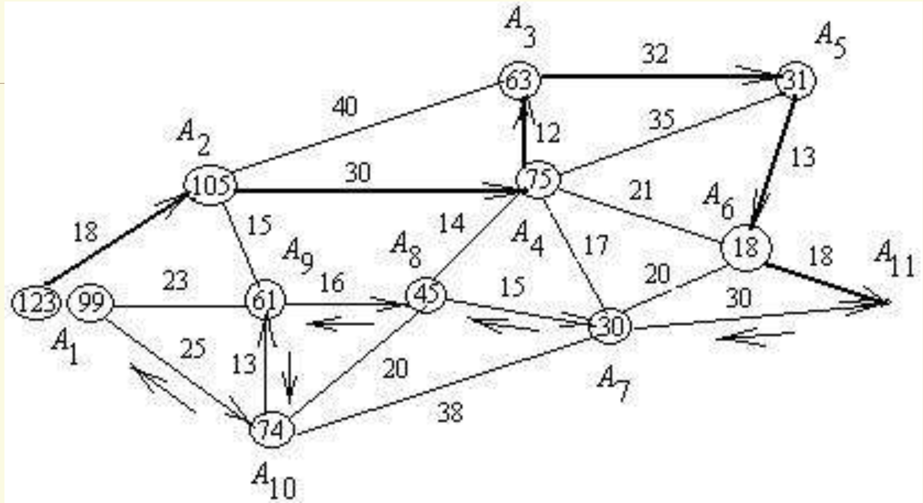
Гамильтонов путь — маршрут, содержащий каждую ***вершину*** графа ровно один раз.

Гамильтонов путь, начальная и конечная вершины которого совпадают, называется **гамильтоновым ци клом**.

Гамильтоновы путь, цикл и граф названы в честь ирландского математика У. Гамильтона, который впервые определил эти классы, исследовав задачу «кругосветного путешествия» по додекаэдру, узловые вершины которого символизировали крупнейшие города Земли, а рёбра — соединяющие их дороги.

ВЗВЕШЕННЫЙ ГРАФ (СЕТЬ)

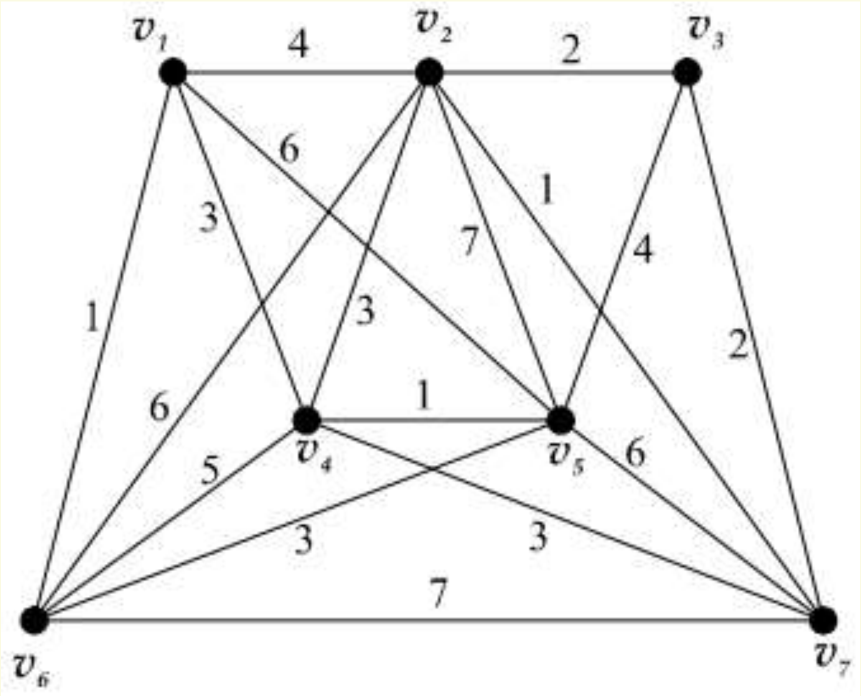
Сетью называется граф, элементам которого поставлены в соответствие некоторые параметры (стоимость, расстояние, пропускная способность и т.п.).



Сеть автомобильных дорог

Построение остовного дерева

В семь населенных пунктов нужно провести кабельное телевидение. Определить маршруты прокладки кабелей вдоль существующих дорог, схема которых представлена на графе, таким образом, чтобы общая длина была минимальной.



Математическая постановка

Предположим, что задан связный граф $<M,N>$, и каждой его дуге $j \in N$ сопоставлено некоторое число $w(j)$, называемое весом или длиной этой дуги.

Сумму весов дуг дерева называют весом дерева.

Требуется найти такое основное дерево, у которого вес был бы минимален.

Такое дерево называют минимальным остовным деревом.

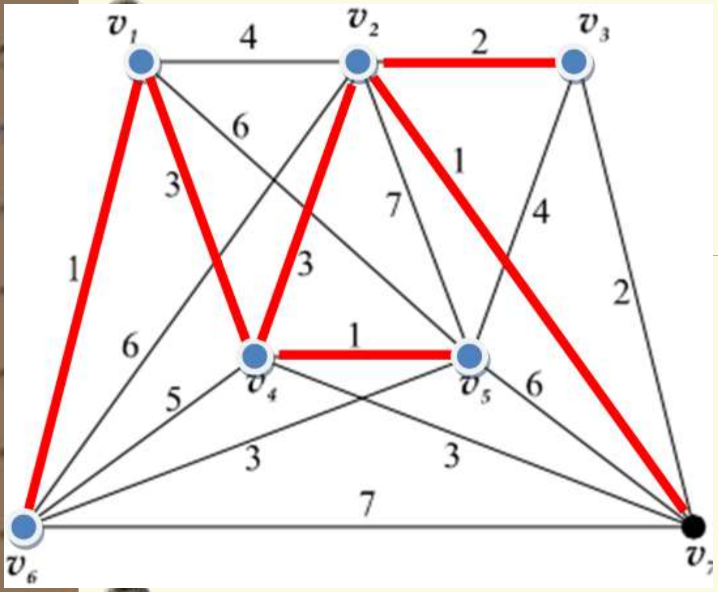
Рассмотрим два метода: алгоритм Прима и алгоритм Краскала

Алгоритм Прима

—алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа.

Алгоритм впервые был открыт в 1930 году чешским математиком Войцехом Ярником, позже переоткрыт Робертом Примом в 1957 году, и, независимо от них, Э.Дейкстрой в 1959 году.

Суть алгоритма: находим ребро минимального веса и выделяем подмножество V* связывающих его вершин. Для всех вершин выделенного подмножества находим ребро с минимальным весом среди всех ребер, ведущих к вершинам, не входящим пока в подмножество V*. Включаем соответствующую вершину в подмножество V*. И так далее, пока все вершины графа не будут включены в подмножествоV*



Алгоритм Прима

Находим ребро минимального веса $(V_1, V_6) = \mathbf{1}$.
Вводим эти вершины в множество $V^*=\{V_1, V_6\}$.
Выбираем ребро минимального веса, исходящее из вершин V*: $(V_1, V_4)=\mathbf{3}$.
Добавляем V_4 в V*: $V^*=\{V_1, V_4, V_6\}$

Выбираем ребро минимального веса, смежное с вершинами V*: $(V_4, V_5)=\mathbf{1}$; добавляем вершину V_5 в V*: $V^*=\{V_1, V_4, V_5, V_6\}$
Выбираем ребро минимального веса, смежное с вершинами V*: $(V_4, V_2) = \mathbf{3}$; добавляем вершину V_2 в V*: $V^*=\{V_1, V_2, V_4, V_5, V_6\}$.
Выбираем ребро минимального веса, смежное с вершинами V*: $(V_2, V_7) = \mathbf{1}$; добавляем вершину V_7 в V*: $V^*=\{V_1, V_2, V_4, V_5, V_6, V_7\}$.
Выбираем ребро минимального веса, смежное с вершинами V*: $(V_2, V_3) = \mathbf{2}$; добавляем вершину V_3 в V*: $V^*=\{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$.
Неохваченных вершин не осталось.

Длина остовного дерева: $\boldsymbol{L}= 1+3+1+3+1+2 = \boldsymbol{11}$

Алгоритм Краскала

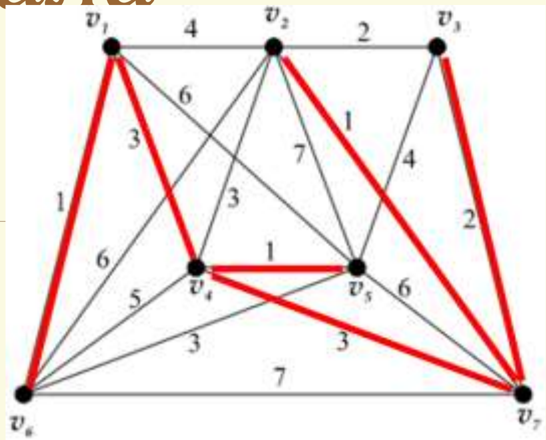
Алгоритм, более привлекательный в вычислительном отношении, был предложен Джозефом Краскалом в 1957 г.

Алгоритм состоит из двух фаз.

Джозеф Краскал

- На подготовительной фазе все дуги удаляются из дерева и упорядочиваются по возрастанию их весов. В графе остаются только вершины, каждая из которых образует отдельную компоненту связности.
- Во второй фазе дуги перебираются в порядке возрастания веса. Если начало и конец очередной дуги принадлежат одной и той же компоненте связности, дуга игнорируется. Если же они лежат в разных компонентах связности. дуга добавляется к графу, а эти две компоненты связности объединяются в одну. Если число компонент связности дойдет до 1, цикл завершается досрочно.

Алгоритм Краскала



Находим ребро минимального веса 1: $(V_1, V_6) = \mathbf{1}$.
Вводим вершины в множество $V^* = \{V_1, V_6\}$
Находим ребро веса 1: $(V_2, V_7) = \mathbf{1}$.
Вводим вершины в множество $V^* = \{V_1, V_2, V_6, V_7\}$
Находим ребро веса 1: $(V_4, V_5) = \mathbf{1}$.
Вводим вершины в множество $V^* = \{V_1, V_2, V_4, V_5, V_6, V_7\}$

Ребер веса 1 больше нет.

Теперь вводим ребра веса 2, так, *чтобы не образовать циклы, но повышать связность графа*.
Вводим в дерево ребро веса 2: $(V_3, V_7) = \mathbf{2}$.
Вводим вершины в множество $V^* = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}$

Ребер веса 2 (не образующих циклов с существующими) больше нет.

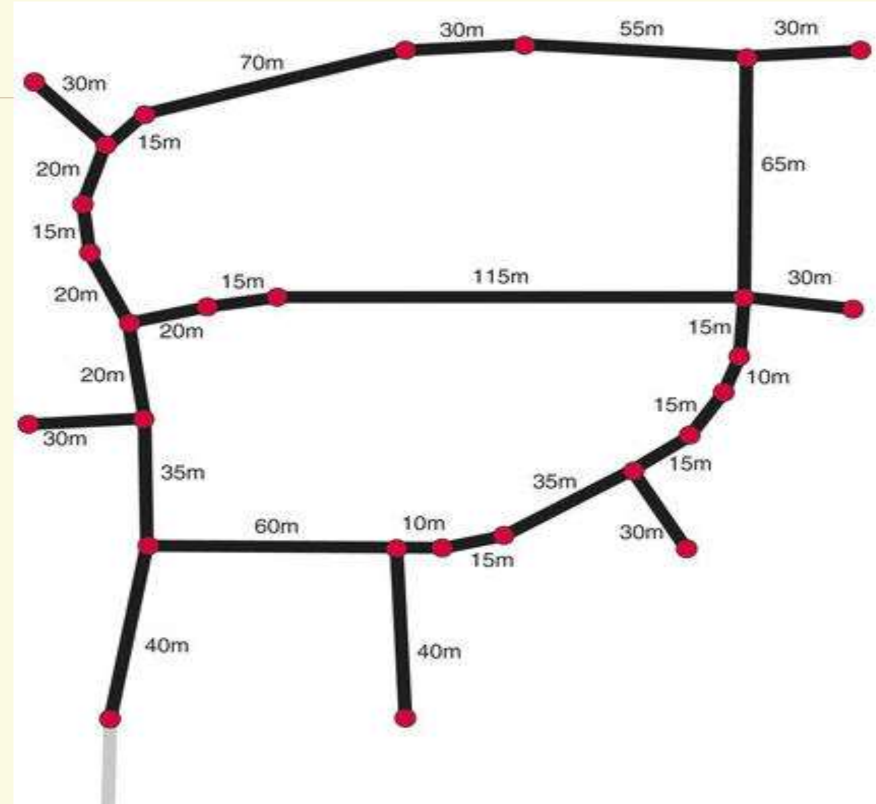
Теперь вводим ребра веса 3, так, *чтобы не образовать циклы, но повышать связность графа.*
Находим ребро веса 3: $(V_1, V_4) = \mathbf{3}$.
Находим ребро веса 3: $(V_4, V_7) = \mathbf{3}$.

Все вершины включены в дерево. Граф связный.

Вес дерева $\boldsymbol{L} = 1+1+1+2+3+3 = \boldsymbol{11}$.

Поиск кратчайшего пути в графе

У нас есть две точки: начало и конец маршрута. Также у нас есть карта, представляющая собой большой массив векторов. Мы должны использовать его и рассчитать оптимальный маршрут.



Существует несколько методов поиска кратчайших путей в графе:

- алгоритм Дейкстры (поиск кратчайших путей *от заданной* вершины);
- алгоритм Флойда (поиска длин *всех* кратчайших путей в графе);
- алгоритм Данцига.

Алгоритм Дейкстры

Метод считается одним из наиболее эффективных алгоритмов решения задачи. Предложен в 1959 г. голландским математиком Дейкстрой (1930-2002), известным, в частности, своей борьбой против безбрежного использования в программировании оператора ***go to***. Эта борьба привела к существенному улучшению стиля программирования.

Эдсгер Вибе **Дейкстра**

Главная идея, лежащая в основе алгоритма Дейкстры

Предположим, что нам известны m вершин, ближайших к вершине **S** (близость любой вершины **X** к вершине **S** определяется длиной кратчайшего пути, ведущего из **S** в **X**). Пусть также известны сами кратчайшие пути, соединяющие вершину **S** с выделенными m вершинами). Нужно ввести правило, как может быть определена (m+1)-я ближайшая к **S** вершина.

Идея алгоритма Дейкстры

Окрасим вершину **S** и **m** ближайших к ней вершин.

Построим для каждой неокрашенной вершины **Y** пути, непосредственно соединяющие с помощью дуг (x, y) каждую окрашенную вершину **X** с **Y**.

Выберем из этих путей кратчайший, и будем считать его условно кратчайшим путем из вершины **S** в вершину **Y**.

Какая же из неокрашенных вершин является (m+1)-й ближайшей к **S** вершиной?

Та, для которой условно кратчайший путь имеет наименьшую длину. Это обусловливается тем, что кратчайший путь из вершины **S** в (m+1)-ю ближайшую вершину при положительном значении длин всех дуг должен содержать в качестве промежуточных лишь окрашенные вершины, т. е. вершины, входящие в число **m** вершин, ближайших к вершине **S**.

Итак, если известны m ближайших к s вершин, то и (m+1)-я ближайшая к **S** вершина может быть найдена.

Начиная с m = 0, описанная процедура может повторяться до тех пор, пока не будет получен кратчайший путь, ведущий из вершины **S** к вершине **T**.

Алгоритм Дейкстры

1) Каждой вершине X в ходе алгоритма присваивается число d(x), равное длине кратчайшего пути из вершины S в вершину X и включающем только окрашенные вершины.
2) Положиv d(s)=0 и d(x)=∞ для всех остальных вершин графа.
3) Окрашиваем вершину S и полагаем Y=S, где Y – последняя окрашенная вершина.
4) Для каждой неокрашенной вершины X пересчитывается величина d(x) по следующей формуле:

$$\mathbf{d(x) = \min \{d(x); d(y) + a_{y,x}\}}$$

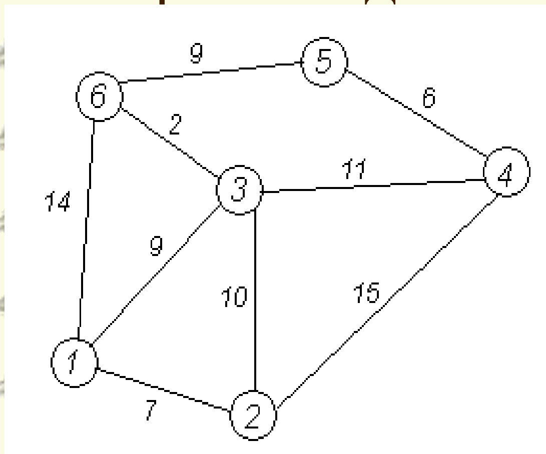
5) Если d(x)=∞ для всех неокрашенных вершин, то алгоритм заканчивается т. к. отсутствуют пути из вершины S в неокрашенные вершины. Иначе окрашивается та вершина, для которой величина d(x) является минимальной.

Окрашивается и дуга, ведущая в эту вершину, и полагаем Y=X.

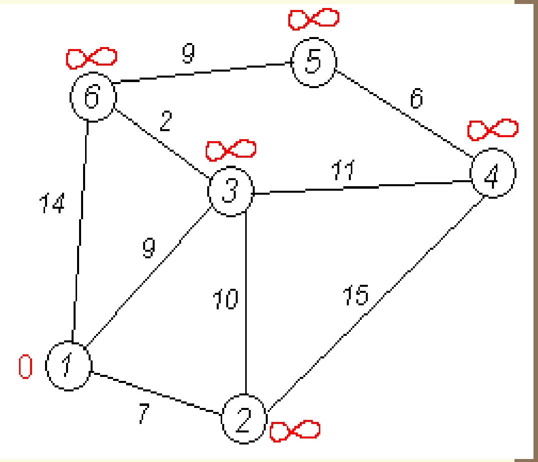
6) Если Y=T, кратчайший путь из S в T найден. Иначе переходим к шагу 2.

ПРИМЕР

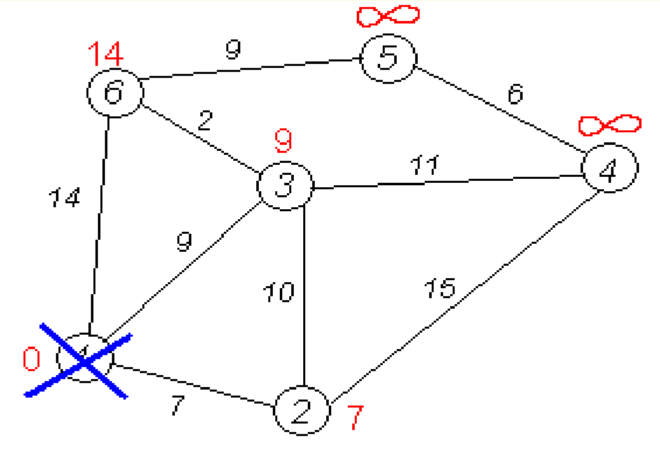
Пусть требуется найти кратчайшие расстояния от 1-й вершины до всех остальных.



Первый шаг. Минимальную метку имеет вершина 1. Её соседями являются вершины 2, 3 и 6.

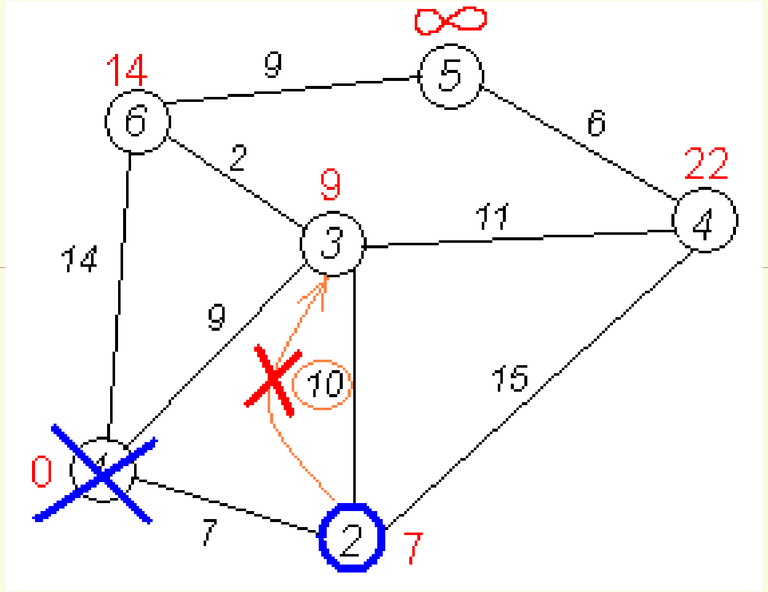


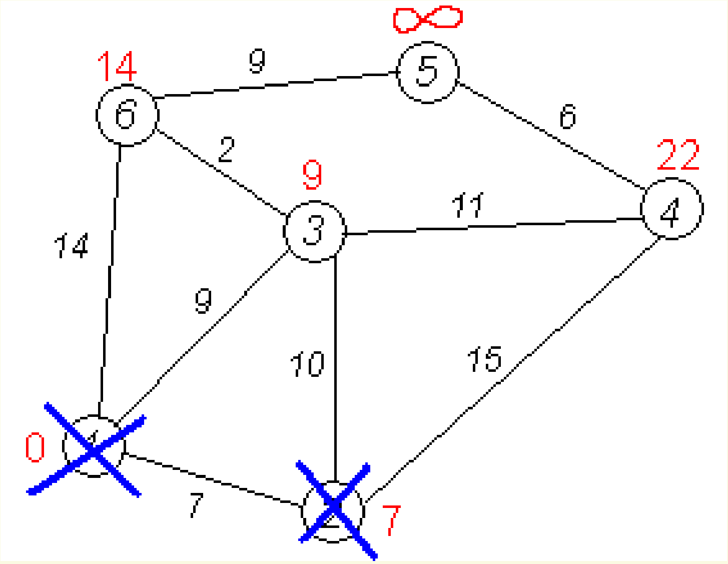
«Проверим» всех «соседей» вершины 1. Текущее минимальное расстояние до вершины 1 считается окончательным и пересмотру не подлежит. Вычеркнем её из графа, чтобы отметить, что эта вершина посещена.



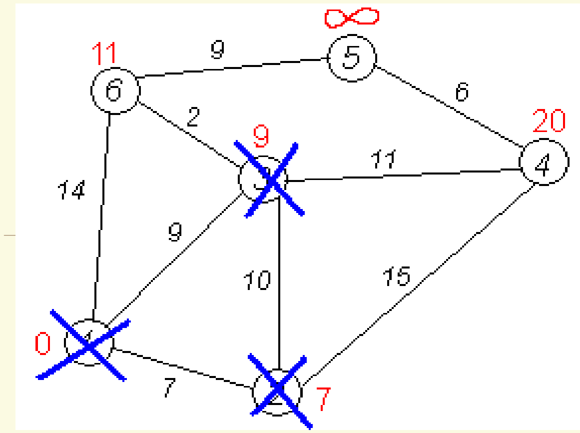
Второй шаг. Шаг алгоритма повторяется. Снова находим «ближайшую» из непосещенных вершин. Это вершина 2 с меткой 7.

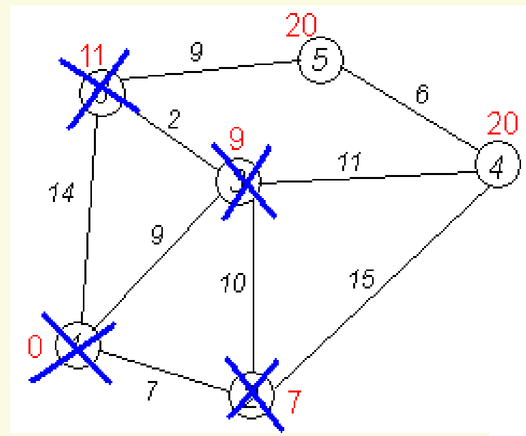
Пытаемся уменьшить метки соседей выбранной вершины, пытаясь пройти в них через 2-ю вершину. Соседями вершины 2 являются вершины 1, 3 и 4.



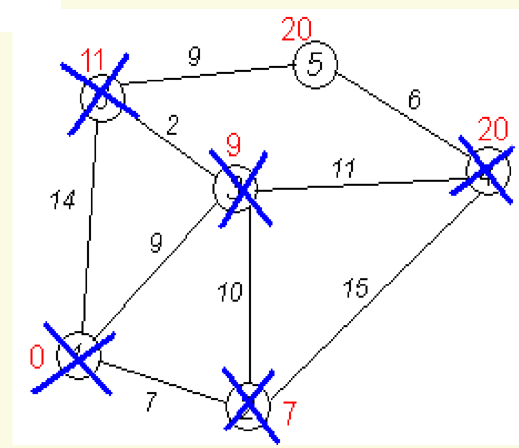


Третий шаг. Повторяем шаг алгоритма, выбрав вершину 3. После её «обработки» получим:





Дальнейшие шаги. Повторяем шаг алгоритма для оставшихся вершин. Это будут вершины 6, 4 и 5, соответственно порядку.



Способы задания графа. Изоморфные графы

Геометрическое представление плоского графа называется его **реализацией**.

При переходе от алгебраического способа (способа задания графа дугами путём указания вершин, которым они инцидентны) к геометрическому одному и тому же графу могут соответствовать различные изображения – **изоморфные графы**.

Граф можно задавать **таблицей,** состоящей из n строк (вершины) и m столбцов (рёбра).

Одним из самых распространённых способов задания графа является **матричный способ**.

Матрицей инцидентности называется таблица В, состоящая из *n* строк (вершины) и *m* столбцов (рёбра), в которой:

- для неориентированного графа:

$b_{ij} = 1$, если вершина V_i инцидентна ребру X_j;

$b_{ij} = 0$, если вершина не инцидентна ребру ;

- для ориентированного графа:

$b_{ij} = 1$, если вершина является началом дуги ;

$b_{ij} = 0$, если вершина не инцидентна дуге ;

$b_{ij} = -1$, если вершина является концом дуги.

Матрицей смежности графа $G = (V, X)$

без кратных рёбер называется квадратная матрица A порядка n, в которой:

$a_{ij} = 1$, если $(V_i, V_j) \in X$;

$a_{ij} = 0$, если $(V_i, V_j) \notin X$.

При восстановлении графа по его матрице инцидентности можно получить граф лишь с *точностью до изоморфизма*.

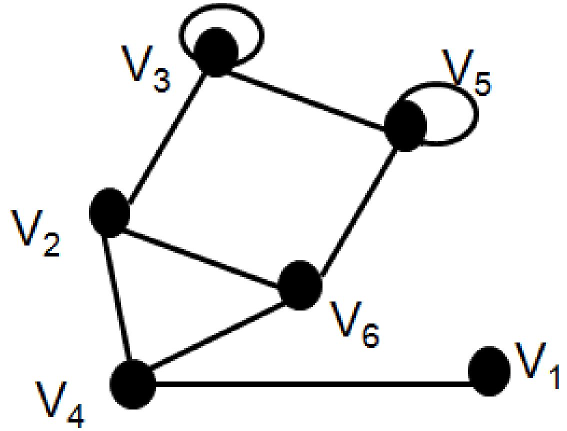
Пример 1

Задача. Пусть граф G задан матрицей смежности A. Построить диаграмму этого графа, если

$$A=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Решение. Диаграмму графа, имеющего шесть вершин, можно представить следующим образом

$$B=\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 1 & \dots & 1 \end{pmatrix}$$



ПРИМЕР 2

Задача. Пусть граф G задан матрицей смежности A. Построить диаграмму этого графа, если

$$A=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Решение. Диаграмму графа, имеющего шесть вершин, можно представить следующим образом

$$B=\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 1 & \dots & 1 \end{pmatrix}$$