

Введение в дискретную математику и математическую логику

•

Лекция №10

Эйлеровы и гамильтоновы пути и циклы

- Апанович Зинаида Владимировна

Часть 1. Эйлеровы пути и циклы

Введение

Можно ли перемещаться по ребрам графа так что, начав с некоторой вершины возвратиться в нее, пройдя **по каждому ребру** графа ровно один раз?

Аналогично, можно ли перемещаться по ребрам графа так что, начав с некоторой вершины возвратиться в нее, посетив **каждую вершину** графа ровно один раз?

Хотя эти вопросы кажутся похожими, НО

на **первый вопрос**, имеет ли график **эйлеров цикл**, можно легко ответить, просто рассмотрев степени вершин графа.

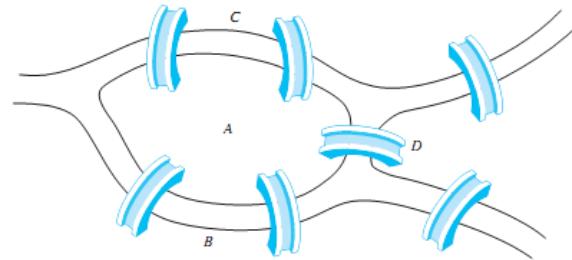
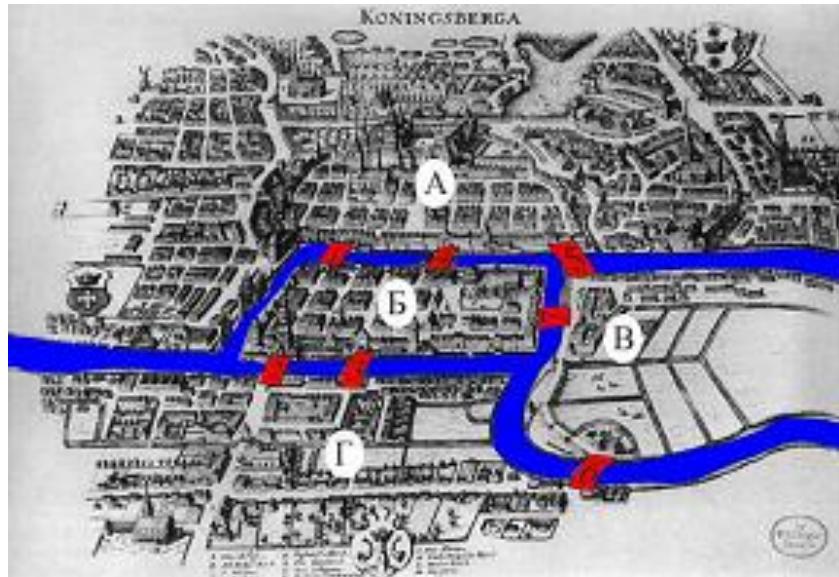
На **второй вопрос**, имеет ли график **гамильтонов цикл**, довольно сложно ответить для большинства графов.

Эйлеровы циклы

Город Кёнигсберг, Пруссия (ныне Калининград, часть Российской Федерации), был разделен на 4 участка рукавами реки Прегель (Преголя) .

Эти 4 участка включали 2 региона на берегах Прегеля, остров Кнайпхоф и регион между двумя рукавами Прегеля.

В XVIII веке эти регионы соединяли 7 мостов.



Эйлеровы циклы

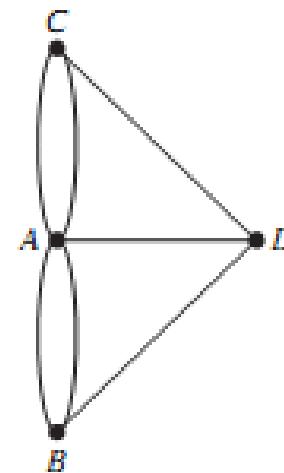
Вопрос: Можно ли начать в каком-либо месте города, проехать по всем мостам один раз, не пересекая ни один мост дважды, и вернуться в исходную точку.

Эту задачу решил швейцарский математик Леонард Эйлер.

Его решение, опубликованное в 1736 году, может быть первым применением теории графов.

Эйлер изучал эту проблему, используя мультиграф, в котором 4 области представлены вершинами, а мосты — ребрами.

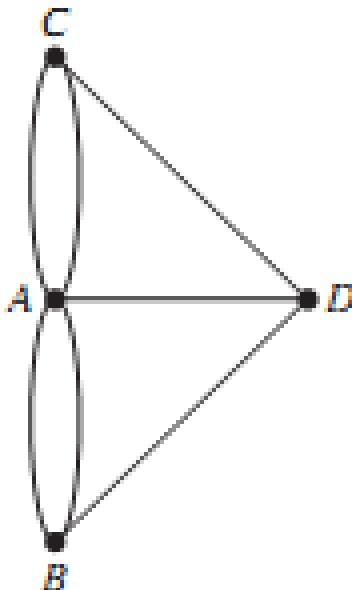
Представление проблемы о мостах Кенигсберга при помощи мультиграфа (Болл, 1892).



Эйлеровы циклы

Проблему проезда по всем мостам, не проходя ни по одному мосту более одного раза, можно перефразировать в терминах этой модели.

Возникает вопрос: существует ли в этом мультиграфе **простой цикл, содержащий все ребра?**



Эйлеровы циклы

ОПРЕДЕЛЕНИЕ 1

Эйлеров цикл в графе G — это простой цикл, содержащий каждое ребро графа G .

Эйлеров путь в графе G — это простой путь , содержащий каждое ребро графа G .

Эйлеровы циклы

ПРИМЕР 1. Какой из неориентированных графов, показанных ниже, имеет эйлеров цикл?

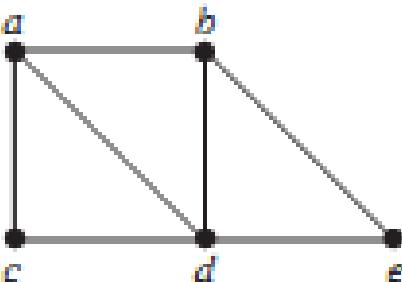
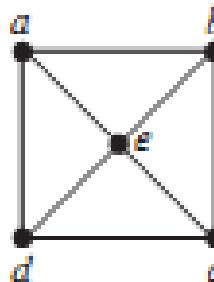
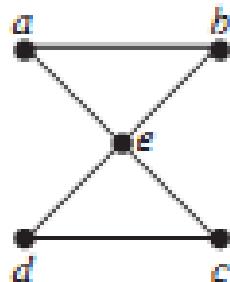
Из тех, которые не имеют цикла, какие имеют эйлеров путь ?

Решение : Граф G_1 имеет эйлеров цикл, например, a, e, c, d, e, b, a .

Ни один из графов G_2 или G_3 не имеет эйлерова цикла.

Однако G_3 имеет эйлеров путь, а именно , a, c, d, e, b, d, a, b .

В G_2 нет эйлерова пути.



G_1

G_2

G_3

Эйлеровы циклы

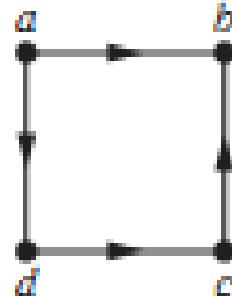
ПРИМЕР 2. Какой из ориентированных графов, показанных ниже, имеет эйлеров цикл?

Какие из них имеют эйлеров путь?

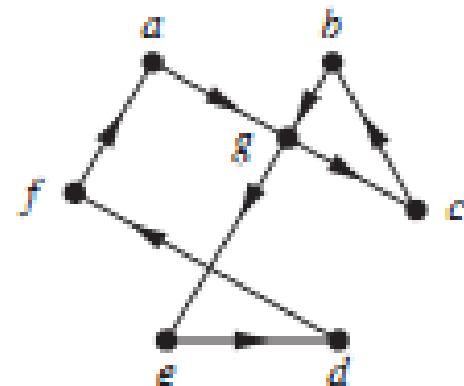
Решение: Граф H_2 имеет эйлеров цикл, например, $a, g, c, b, g, e, d, f, a$.

Ни H_1 ни H_3 не имеют эйлерова цикла.

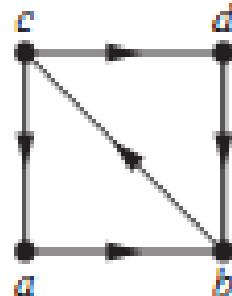
H_3 имеет эйлеров путь, а именно c, a, b, c, d, b , но в H_1 его нет.



H_1



H_2



H_3

Необходимые и достаточные условия существования эйлеровых циклов

Существуют простые критерии для определения того, имеет ли мультиграф **эйлеров цикл** или **эйлеров путь**.

Мы будем предполагать, что все графы, обсуждаемые в этом разделе, имеют **конечное число вершин и ребер**.

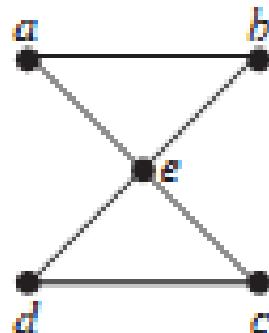
Что можно сказать, если связный **мультиграф** имеет эйлеров цикл?

Мы можем показать, что **каждая вершина такого мультиграфа должна иметь четную степень**.

Для этого сначала отметим, что эйлеров цикл начинается с вершины a и продолжается ребром, инцидентным a , скажем, $\{a, b\}$.

Ребро $\{a, b\}$ вносит вклад 1 в $\deg(a)$.

Каждый раз, когда цикл проходит через вершину, он увеличивает **степень вершины на 2**, поскольку цикл входит через ребро, инцидентное этой вершине, и выходит через другое такое же ребро.



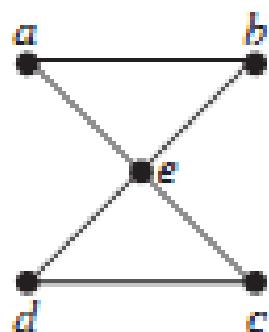
Необходимые и достаточные условия существования эйлеровых циклов

Наконец, цикл завершается [там, где он начался](#), внося 1 в $\deg(a)$.

Следовательно, $\deg(a)$ должна быть [четной](#), поскольку цикл вносит 1, когда он начинается, 1, когда он заканчивается, и 2 каждый раз, когда он проходит через a (если он когда-либо это делает).

Вершина, отличная от a , имеет [четную](#) степень, поскольку цикл увеличивает ее степень на 2 каждый раз, когда проходит через вершину.

Мы приходим к выводу, что если связный граф имеет эйлеров цикл, то каждая вершина должна иметь [четную степень](#).



Необходимые и достаточные условия существования эйлеровых циклов

Является ли это **необходимое** условие существования эйлерового цикла также **достаточным**?

То есть, должен ли существовать эйлеров цикл в связном мультиграфе, **если все вершины имеют четную степень?**

Этот вопрос можно решить **утвердительно** с помощью конструктивного построения.

Предположим, что G — связный **мультиграф, имеющий** не менее чем **2** вершины и при этом степень каждой вершины G четна.

Мы сформируем простой цикл, который начинается в **произвольной вершине a** графа G , проходя **ребро за ребром**.

Пусть $x_0 = a$.

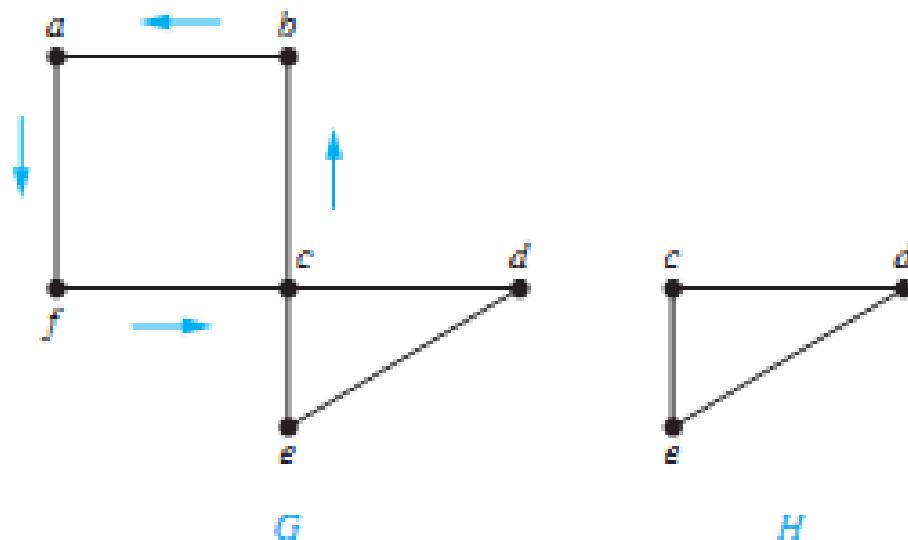
Сначала мы произвольно выбираем ребро $\{x_0, x_1\}$, инцидентное вершине a , это возможно, поскольку G связан.

Продолжаем строить простой путь $\{x_0, x_1\}, \{x_1, x_2\}, \dots, \{x_{n-1}, x_n\}$, последовательно добавляя ребра одно за другим к пути до тех пор, пока мы больше не сможем добавить еще одно ребро к пути.

Необходимые и достаточные условия существования эйлеровых циклов

Это происходит, когда мы достигаем вершины, для которой мы уже включили в путь все ребра, инцидентные этой вершине.

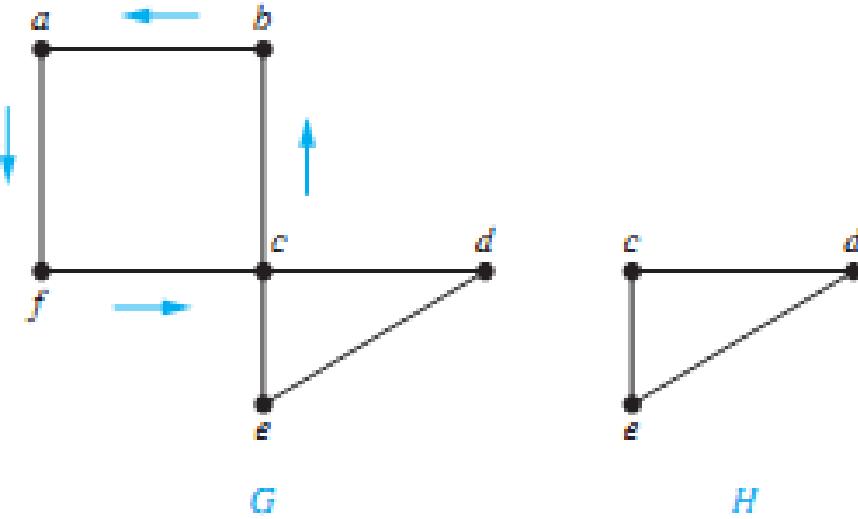
Например, в графе G , показанном ниже, мы начинаем с вершины a и последовательно выбираем ребра $\{a, f\}$, $\{f, c\}$, $\{c, b\}$ и $\{b, a\}$.



Необходимые и достаточные условия существования эйлеровых циклов

Построенный нами путь **должен закончиться**, поскольку граф имеет **конечное** количество ребер.

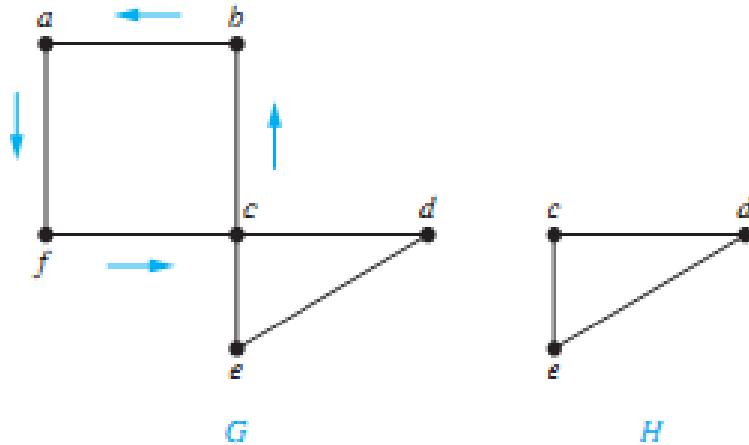
Поэтому мы гарантированно в конечном итоге достигнем вершины, для которой не будет ребер, доступных для добавления к пути.



Необходимые и достаточные условия существования эйлеровых циклов

1) Путь начинается в **вершине a** ребром, имеющим вид $\{a, x\}$, и теперь мы покажем, что он должен заканчиваться в **вершине a** ребром, имеющим вид $\{y, a\}$.

Чтобы понять, что путь должен заканчиваться в **вершине a** , заметим, что каждый раз, когда путь проходит через вершину с **четной степенью**, он использует только **1 ребро для входа в эту вершину**, поэтому, поскольку степень должна быть не менее 2, для выхода пути из вершины остается не менее **1 ребра**.

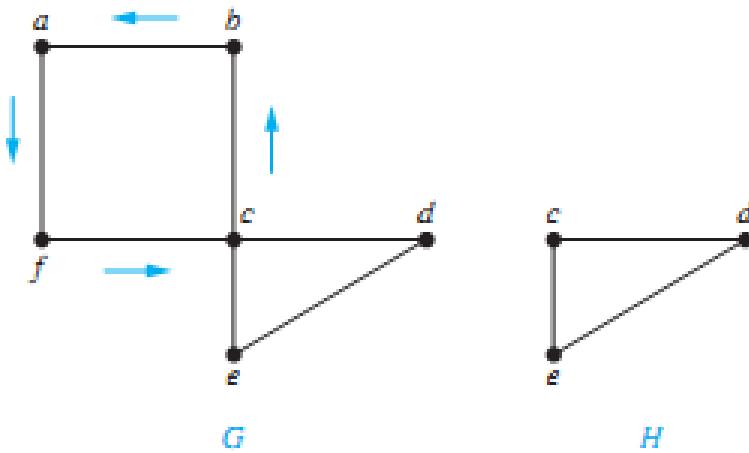


Необходимые и достаточные условия существования эйлеровых циклов

Более того, каждый раз, когда мы входим и выходим из вершины **четной степени**, остается **четное число ребер**, инцидентных этой вершине, которые мы еще **не использовали** на нашем пути.

Следовательно, при формировании пути каждый раз, когда мы входим в вершину, отличную от a , мы можем ее покинуть.

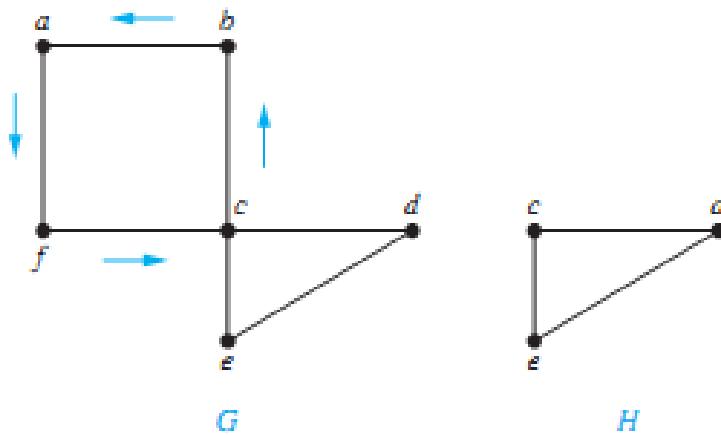
Это означает, что путь может заканчиваться только в вершине a .



Необходимые и достаточные условия существования эйлеровых циклов

2) Обратите внимание, что построенный нами путь **может либо использовать все ребра графа**, либо это **может быть не так**, если мы вернулись вершину a в последний раз **перед использованием всех ребер**.

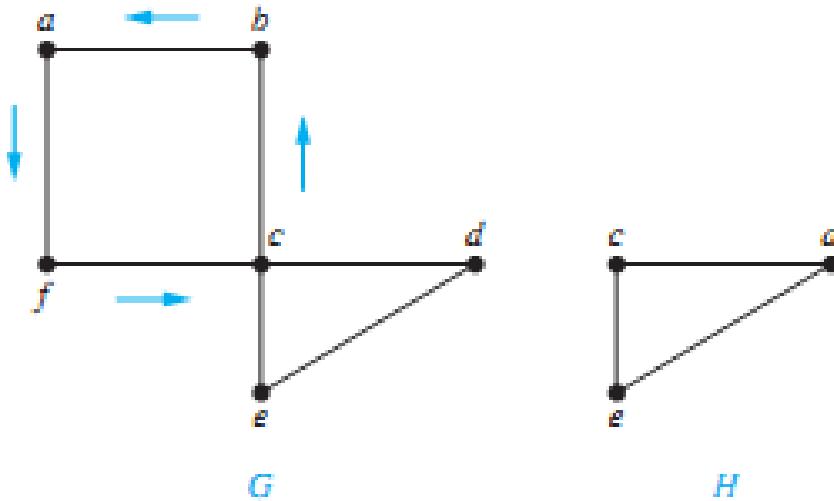
Если все ребра были использованы, то будет построен эйлеров цикл.



Необходимые и достаточные условия существования эйлеровых циклов

В противном случае рассмотрим подграф H , полученный из G путем удаления уже использованных ребер и вершин, не инцидентных ни одному из оставшихся ребер.

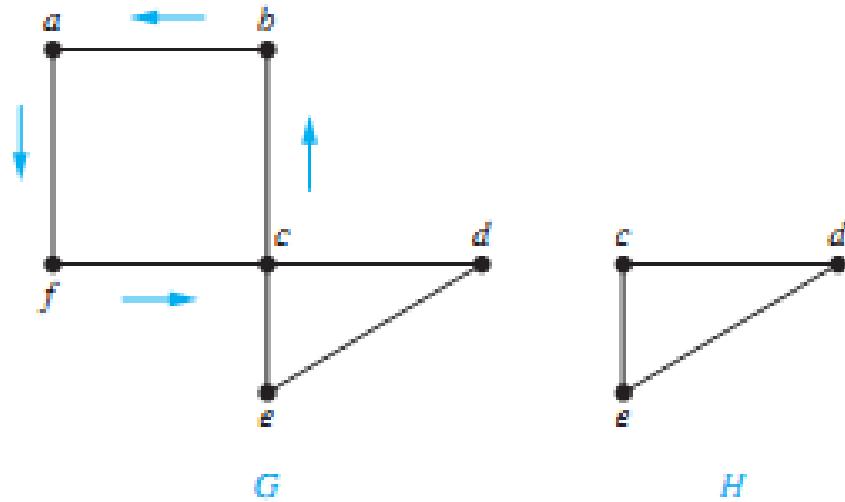
Удалив цикл a, f, c, b, a из графа G , мы получим подграф, обозначенный как H .



Необходимые и достаточные условия существования эйлеровых циклов

Поскольку G связан, H имеет по крайней мере одну общую вершину с удаленным циклом.

Пусть w будет такой вершиной. (В нашем примере такой вершиной является c .)



Необходимые и достаточные условия существования эйлеровых циклов

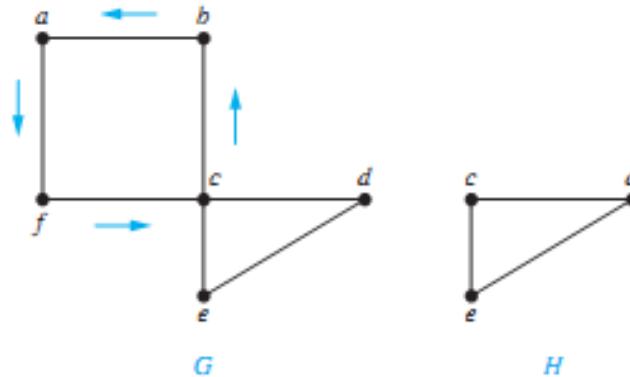
Каждая вершина в H имеет **четную степень** (потому что в G все вершины имели четную степень, и для каждой вершины пары ребер, инцидентных этой вершине, были удалены, чтобы сформировать H).

Обратите внимание, что H может быть не связен.

Начиная с w , построим простой путь как можно большей длины в H , выбирая ребра, как это было сделано в G .

Этот путь должен заканчиваться в вершине w .

Например, c, d, e, c — это путь в H .



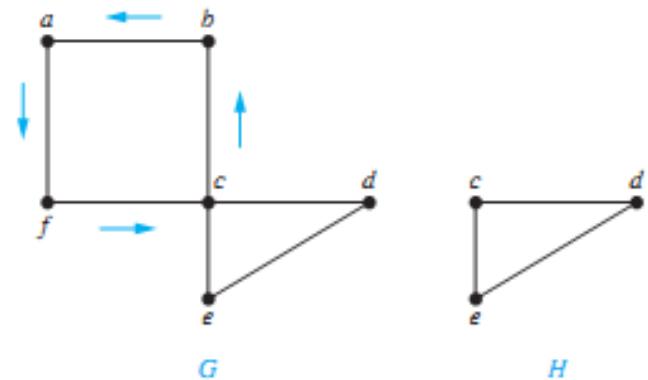
Необходимые и достаточные условия существования эйлеровых циклов

Затем сформируем цикл в G , соединив цикл в H с исходным циклом в G . (это можно сделать, поскольку w является одной из вершин в этого цикла).

Если это сделать на графике ниже, то получим цикл a, f, c, d, e, c, b, a . Продолжаем этот процесс, пока все ребра не будут использованы. (Процесс должен завершиться, поскольку в графике имеется только конечное число ребер.)

Это создает эйлеров цикл.

Построение показывает, что если все вершины связного мультиграфа имеют четную степень, то график имеет эйлеров цикл.



Необходимые и достаточные условия существования эйлеровых циклов

ТЕОРЕМА 1 Связный мультиграф, имеющий
не менее 2 вершин, имеет эйлеров цикл \Leftrightarrow
каждая его вершина имеет **четную** степень.

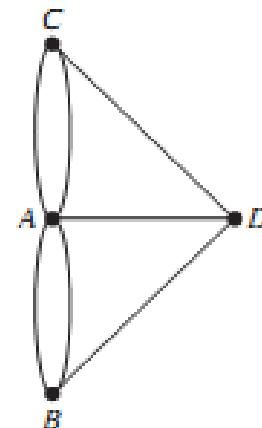
Необходимые и достаточные условия существования эйлеровых циклов

Теперь мы можем решить задачу о мостах Кенигсберга.

Поскольку мультиграф, представляющий эти мосты, имеет 4 вершины нечетной степени, он не имеет эйлерова цикла.

Невозможно начать в определенной вершине, пройти по каждому мосту ровно один раз и вернуться в исходную вершину.

Алгоритм 1 дает конструктивную процедуру нахождения эйлеровых циклов, приведенную в обсуждении, предшествующем теореме 1



Необходимые и достаточные условия существования эйлеровых циклов

ALGORITHM 1 Constructing Euler Circuits.

procedure *Euler*(*G*: connected multigraph with all vertices of even degree)

circuit := a circuit in *G* beginning at an arbitrarily chosen vertex with
 edges successively added to form a path that
 returns to this vertex

H := *G* with the edges of this *circuit* removed

while *H* has edges

subcircuit := a circuit in *H* beginning at a vertex in *H* that also is an
 endpoint of an edge of *circuit*

H := *H* with edges of *subcircuit* and all isolated vertices removed

circuit := *circuit* with *subcircuit* inserted at the appropriate vertex

return *circuit* {*circuit* is an Euler circuit}

Необходимые и достаточные условия существования эйлеровых циклов

Алгоритм 1 обеспечивает эффективный алгоритм нахождения эйлеровых циклов в связном мультиграфе G , имеющем все вершины четной степени.

Сложность этого алгоритма в худшем случае составляет $O(m)$, где m — количество ребер графа G .

Необходимые и достаточные условия существования эйлеровых циклов

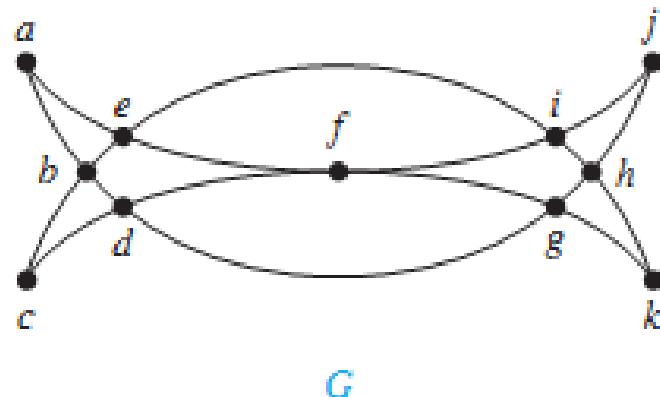
ПРИМЕР 3. Можно ли нарисовать сабли Магомета, показанные ниже, таким образом, чтобы рисунок начинался и заканчивался в одной и той же вершине?

Решение : Мы можем решить эту задачу, поскольку граф G имеет эйлеров цикл.

Он имеет такой цикл, поскольку все его вершины имеют **четную степень**.

Мы будем использовать Алгоритм1 для построения эйлерова цикла.

Сначала формируем цикл $a, b, d, c, b, e, i, f, e, a$. Получаем подграф H , удаляя ребра этого цикла и все вершины, которые становятся изолированными при удалении этих ребер.



Сабли Магомета.

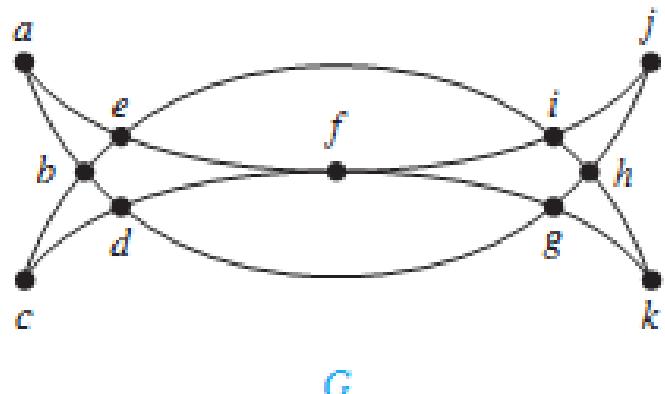
Необходимые и достаточные условия существования эйлеровых циклов

Затем формируем цикл $d, g, h, j, i, h, k, g, f, d$ в H .

Для этого цикла мы использовали все ребра в G .

Встраивание этого нового цикла в первый цикл в соответствующем месте дает Эйлеров цикл $a, b, d, g, h, j, i, h, k, g, f, d, c, b, e, i, f, e, a$.

Эта схема позволяет рисовать сабли, не отрывая карандаша от бумаги и не пересыпывая часть рисунка.



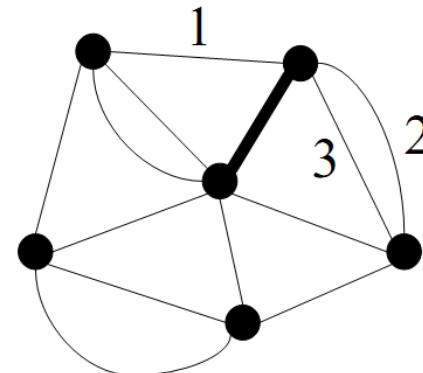
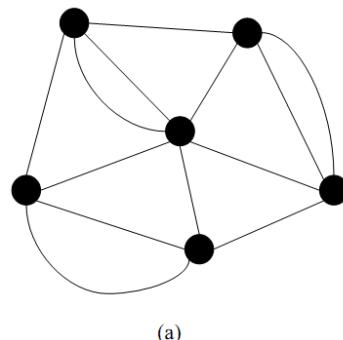
Алгоритм Флери построения Эйлерова цикла

Другой алгоритм построения Эйлерова цикла, называемый [алгоритмом Флери](#), был опубликован в 1883 году .

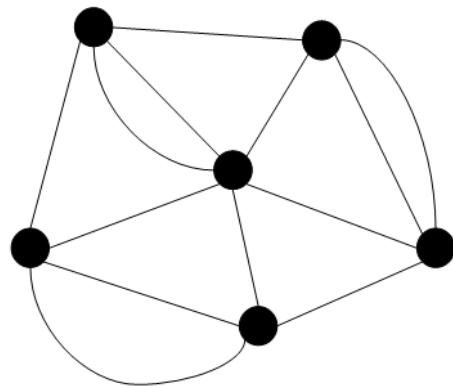
Алгоритм Флери строит эйлеров цикл, сначала выбирая произвольную вершину связного мультиграфа, а затем формируя цикл путем последовательного выбора ребер.

После выбора ребра, оно удаляется.

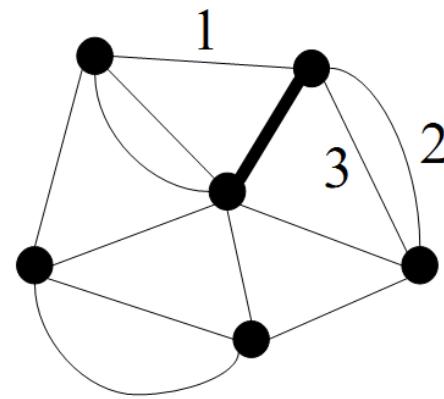
Ребра выбираются последовательно так, чтобы каждое ребро начиналось там, где кончается предыдущее ребро, и так, чтобы выбранное ребро [не было мостом, кроме случаев](#), когда нет альтернатив.



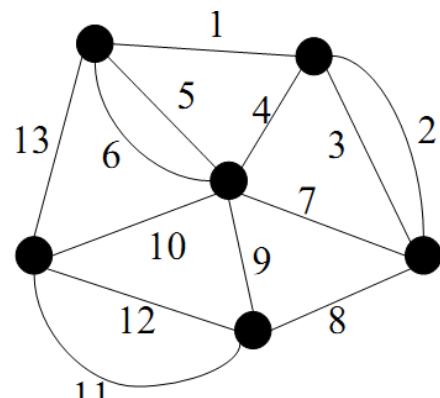
Алгоритм Флери построения Эйлерова цикла



(a)



(b)



(c)

Необходимые и достаточные условия существования эйлеровых путей

Теперь покажем, что связный мультиграф имеет эйлеров путь (а не эйлеров цикл) \Leftrightarrow он имеет ровно 2 вершины нечетной степени.

=> Предположим, что связный мультиграф имеет эйлеров путь из вершины a в вершину b , но не имеет эйлерова цикла.

Первое ребро пути добавляет 1 к степени вершины a .

Вклад 2 в степень a вносится каждый раз, когда путь проходит через a .

Последнее ребро в пути добавляет 1 к степени b .

Каждый раз, когда путь проходит через b , ее степень увеличивается на 2.

Следовательно, и a , и b имеют нечетную степень.

Каждая другая вершина имеет четную степень, поскольку путь добавляет 2 к степени вершины всякий раз, когда он проходит через нее.

Необходимые и достаточные условия существования эйлеровых путей

<= Предположим, что граф имеет ровно 2 вершины нечетной степени, скажем, a и b .

Рассмотрим больший граф, состоящий из исходного графа с добавлением ребра $\{a, b\}$.

Каждая вершина этого большего графа имеет четную степень, поэтому существует эйлеров цикл.

Удаление нового ребра создает эйлеров путь в исходном графе.

Теорема 2 резюмирует эти результаты.

ТЕОРЕМА 2 Связный мультиграф имеет эйлеров путь, но не имеет эйлерова цикла \Leftrightarrow он имеет ровно 2 вершины нечетной степени.

Необходимые и достаточные условия существования эйлеровых путей

ПРИМЕР 4. Какие из представленных ниже графов имеют Эйлеров путь?

Решение : G_1 содержит ровно 2 вершины *нечетной степени*, а именно b и d .

Следовательно, он имеет Эйлеров путь, который должен иметь b и d в качестве своих конечных точек.

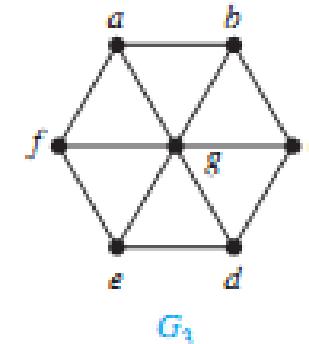
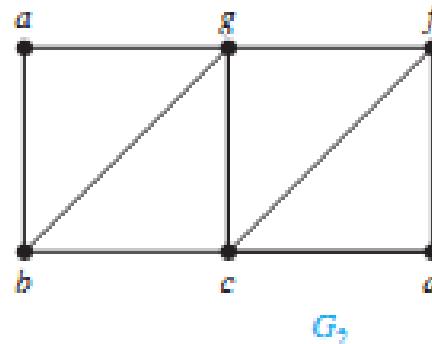
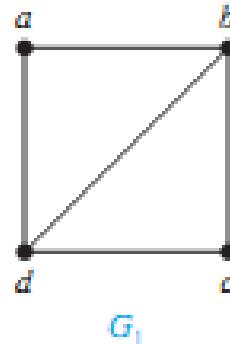
Одним из таких Эйлеровых путей является d, a, b, c, d, b .

Аналогично, G_2 имеет ровно 2 вершины нечетной степени, а именно b и d .

Таким образом, у него есть эйлеров путь, конечными точками которого должны быть b и d .

Одним из таких путей Эйлера является $b, a, g, f, e, d, c, g, b, c, f, d$.

В графе G_3 нет эйлерова пути, поскольку он имеет 6 вершин нечетной

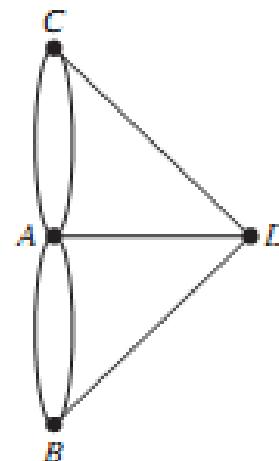


Необходимые и достаточные условия существования эйлеровых путей

Возможно ли начать с какой-то точки Кёнигсберга XVIII века, проехать по всем мостам и оказаться в какой-то другой точке города?

На этот вопрос можно ответить, определив, существует ли **эйлеров путь** в мультиграфе, представляющем мосты Кёнигсберга.

Поскольку в этом мультиграфе **4** вершины **нечетной степени**, эйлерова пути не существует.



Приложения эйлеровых путей и циклов

Эйлеровы пути и циклы можно использовать для решения многих практических задач.

Например, многие приложения требуют найти путь или цикл, который проходит через

- каждую улицу в районе,
- каждую дорогу в транспортной сети,
- или каждое звено в сети связи только один раз.

Нахождение эйлерового пути или цикла в соответствующей графовой модели может решить такие проблемы.

Например, если почтальон может найти эйлеров путь в графе, представляющем улицы, которые ему нужно пройти, этот путь создаст маршрут, который проходит по каждой улице маршрута ровно один раз.

Если эйлерова путь не существует, некоторые улицы придется проходить более одного раза.

Задача нахождения цикла в графе с [наименьшим количеством ребер](#), который проходит через каждое ребро хотя [бы один раз](#), известна как [задача о китайском почтальоне](#) в честь Гуань Мэйгу , который сформулировал ее в 1962 году.

Среди других областей, где применяются эйлеровы циклы и пути, — компоновка схем, сетевая многоадресная передача и молекулярная биология, где эйлеровы пути используются при секвенировании ДНК.

Эйлеровы пути и цепочки РНК

Белки кодируются в живых клетках цепочками рибонуклеиновой кислоты (РНК).

Используются 4 азотистых основания : аденин (A), цитозин (C), гуанин (G) и урацил (U).

Таким образом, цепочку РНК можно обозначить последовательностью букв алфавита A, C, G, U.

Существуют некоторые правила использования этого алфавита:
азотистые основания объединяются в тройки, называемые *кодонами* .

и есть один кодон (**AUG**), который всегда начинает цепочку и несколько других кодонов (**UAA**, **UAG**, **UGA**), которые могут завершить цепочку.

Рассмотрим гипотетическую цепь РНК, которая будет иметь вид:

AUG CAGCCUAUGGGAAAAA UAG .

Эйлеровы пути и цепочки РНК

Чтобы секвенировать цепочку РНК, биологи обрабатывают ее одним или несколькими ферментами, чтобы разбить ее на более короткие фрагменты.

Один из этих ферментов разрывает цепь после каждого **G**, а другой разрывает цепь после каждого **C** и после каждого **U**.

Наша гипотетическая цепочка РНК AUGCAGCCUAUGGGAAAAUAG будет разбита **G-ферментом** на фрагменты AUG, CAG, CCUAUG, G, G, AAAAUAG, а с помощью **CU-фермента** на фрагменты AU, GC, AGC, C, U, AU, GGGAAAAU, AG.

Но, конечно, в реальности эти фрагменты цепочки РНК не будут располагаться в правильном порядке; они будут все перемешаны.

Математическая задача состоит в том, чтобы вывести последовательность цепочки РНК из фрагментов, разрушенных ферментами.

При наивном подсчете, существует $6! = 720$ различных цепочек, которые могли бы соответствовать нашему примеру, нарушенному с помощью G-фермента, и $8! = 40\ 320$ различных цепочек, которые могли бы соответствовать нашему примеру, нарушенному CU-ферментом.

Эйлеровы пути и цепочки РНК

Во-первых, заметим, что можно определить конец цепочки:

В нашем примере один из CU-битов заканчивается на G.

CU-биты: AU, GC, AGC, C, U, AU, GGGAAAAU, **AG**

AG должен быть конечным фрагментом цепочки, поскольку CU-фермент не разорвал его.

G-биты: AUG, CAG, CCUAUG, G, G, AAAAUAG

В более общем случае будет ≤ 1 CU-бита, заканчивающегося на A или G и ≤ 1 G-бита, оканчивающегося на A, C или U.

Также, в худшем случае будет один бит каждого типа, заканчивающегося на A, но поскольку оба бита должны заканчивать цепочку, один из них будет подбитом другого, и все будет хорошо.

Эйлеровы пути и цепочки РНК

G-биты: AUG, CAG, CCUAUG, G, G, AAAAUAG

Теперь мы возьмем каждый G-бит и разобьем его с помощью CU-фермента;

например, CCUAUG становится C, C, U, AU, G.

AUG-> AU, G

CAG -> C, AG

CCUAUG -> C, C, U, AU, G

G

G

AAAAUAG-> AAAAU, AG-

Эйлеровы пути и цепочки РНК

Аналогично мы возьмем каждый CU-бит и разобьем его с помощью G-фермента; например, AGC станет AG, C.

CU-биты: AU, GC, AGC, C, U, AU, GGGAAAAU, **AG**

AC,

GC -> G, C

AGC -> AG, C.

C

U

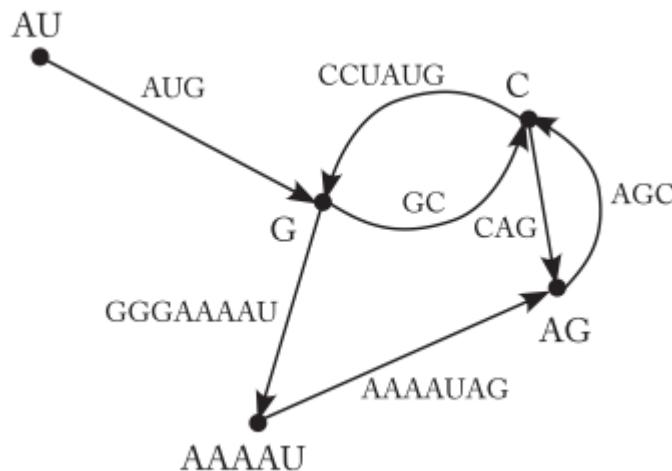
AU

GGGAAAAU -> G, G, G, AAAAU

AG

Эйлеровы пути и цепочки РНК

Теперь создадим ориентированное ребро, соответствующее каждому G-биту и каждому CU-биту, обращая внимание только на первый и последний результирующие биты дважды разорванных последовательностей. То есть первый бит каждого ребра станет отправной вершиной ребра, а последний бит – концевой вершиной. Поэтому примеры из предыдущих двух предложений становятся **C CCUAU G → C, G,**
а **CAG → C, AG**

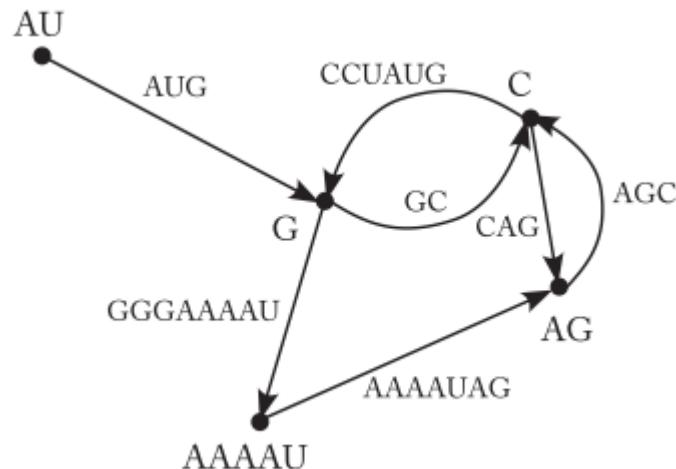


Эйлеровы пути и цепочки РНК

Затем мы склеиваем все эти ребра в ориентированный граф.

Для гипотетической цепочки РНК, которую мы здесь использовали, мы получаем орграф, показанный ниже.

(Обратите внимание, что мы не включаем фрагменты, которые разбиты только один раз, а не два раза.)



Эйлеровы пути и цепочки РНК

Затем мы пытаемся найти эйлеров путь орграфа, который заканчивается известным конечным битом.

Каждый Эйлеров путь можно «прочитать», перечислив метки ребер в том порядке, в котором мы их проходим, исключив первые/последние общие буквы.

Если нам повезет, то будет только один эйлеров путь.

Для нашего примера орграфа существует несколько эйлеровых путей — мы должны начать нашу реконструкцию с AUG, но затем можем продолжить как

AUG GGAAAAUAGCCUAUGC AG ,

как AUGCAGCCUAUGGGAAAAAUAG ,

или как AUG CCUAUGGGAAAAAUAGC AG .

Часть 2

Гамильтоновы циклы и пути

Гамильтоновы пути и циклы

Мы рассмотрели необходимые и достаточные условия существования путей и циклов, содержащих **каждое ребро мультиграфа ровно один раз**.

Можем ли мы сделать то же самое для простых путей и циклов, которые содержат **каждую вершину графа ровно один раз?**

Гамильтоновы пути и циклы

Определение 1. Простой путь в графе G , который проходит через **каждую вершину ровно один раз**, называется **гамильтоновым путем**, а

Простой цикл в графе G , проходящий через **каждую вершину ровно один раз**, называется **гамильтоновым циклом**.

То есть, простой путь $x_0, x_1, \dots, x_{n-1}, x_n$ в графе $G = (V, E)$ является **гамильтоновым путем**, если $V = \{x_0, x_1, \dots, x_{n-1}, x_n\}$ и $x_i \neq x_j$ для $0 \leq i < j \leq n$, и простой цикл $x_0, x_1, \dots, x_{n-1}, x_n, x_0$ (при $n > 0$) является **гамильтоновым циклом**, если $x_0, x_1, \dots, x_{n-1}, x_n$ является **гамильтоновым путем**.

Гамильтоновы пути и циклы

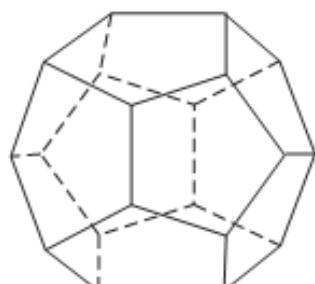
Этот термин происходит от игры под названием « [Икосианская головоломка](#) » , изобретенной в 1857 году ирландским математиком сэром Уильямом Роэнном Гамильтоном.

Он состоял из деревянного додекаэдра [многогранника с 12 правильными пятиугольниками в качестве граней, как показано ниже] со штифтом в каждой вершине додекаэдра и веревки.

20 вершин додекаэдра были помечены названиями разных городов мира.

Цель головоломки состояла в том, чтобы начать с некоторого города и путешествовать по граням додекаэдра, посетив каждый из 19 других городов ровно один раз, и закончить в первом городе.

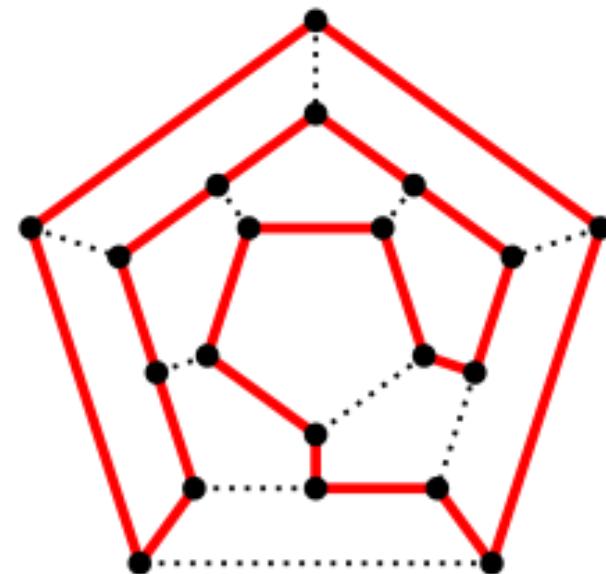
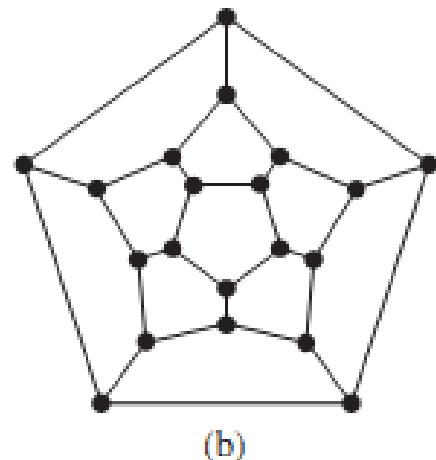
Пройденный путь размечался с помощью веревки и колышков.



Гамильтоновы пути и циклы

Рассмотрим эквивалентный вопрос: существует ли в графе, показанном ниже, цикл, проходящий через каждую вершину ровно один раз?

Поскольку этот граф изоморден графу, состоящему из вершин и ребер додекаэдра, мы получаем решение головоломки.



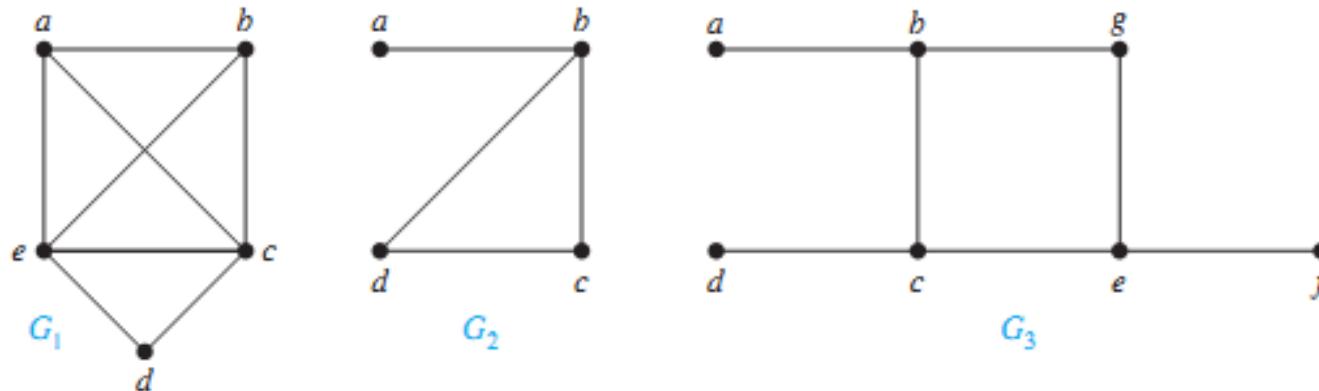
Гамильтоновы пути и циклы.

ПРИМЕР 1. Какие из простых графов, представленных ниже, имеют гамильтонов цикл и, если нет, то гамильтонов путь?

Решение : G_1 имеет гамильтонов цикл : a, b, c, d, e, a .

В графе G_2 нет гамильтонова цикла (это можно увидеть, заметив, что любой цикл, содержащий каждую вершину, должен содержать ребро $\{a, b\}$ дважды), но в графе G_2 есть гамильтонов путь, а именно a, b, c, d .

G_3 не имеет ни гамильтонова цикла, ни гамильтонова пути, поскольку любой путь, содержащий все вершины, должен содержать одно из ребер $\{a, b\}$, $\{e, f\}$ и $\{c, d\}$ более одного раза.



Гамильтоновы пути и циклы.

ПРИМЕР 2. Показать, что K_n имеет гамильтонов цикл для всех $n \geq 3$.

Решение: Мы можем построить цикл в K_n , начинающийся в любой вершине.

Такой цикл можно построить, посещая вершины в любом выбранном нами порядке, при условии, что путь начинается и заканчивается в одной и той же вершине и посещает каждую другую вершину ровно один раз, поскольку в K_n есть ребра между любыми двумя вершинами.

Условия существования гамильтоновых циклов

Существует ли простой способ определить, имеет ли граф гамильтонов цикл или путь?

На первый взгляд может показаться, что должен быть простой способ определить это, поскольку существует простой способ ответить на аналогичный вопрос о том, имеет ли граф эйлеров цикл.

Удивительно, но не существует известных простых необходимых и достаточных критериев существования гамильтоновых циклов.

Однако известно много теорем, дающих достаточные условия доказательства существования гамильтоновых циклов.

Свойства, показывающие, что граф не имеет гамильтонова цикла

Кроме того, некоторые свойства можно использовать для того, чтобы показать, что **граф не имеет гамильтонова цикла**.

- 1) Граф с вершиной **степени 1** не может иметь гамильтонов цикл, потому что в гамильтоновом цикле каждая вершина инцидентна **2** ребрам цикла.
- 2) Если вершина в графе имеет степень **2**, то **оба ребра**, инцидентные этой вершине, должны быть частью любого гамильтонова цикла.
- 3) Когда строится цикл, и этот цикл проходит через некоторую вершину, то все оставшиеся ребра, инцидентные этой вершине, кроме **двух**, используемых в цикле, можно исключить из рассмотрения.
- 4) Гамильтонов цикл не может содержать внутри себя цикл меньшего размера.

Гамильтоновы циклы

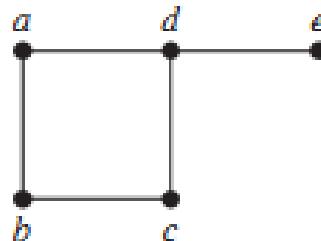
ПРИМЕР 3. Показать, что ни один из представленных ниже графов не содержит гамильтонова цикла.

Решение: В графе G нет цикла, поскольку в графе G есть вершина степени 1, а именно, e .

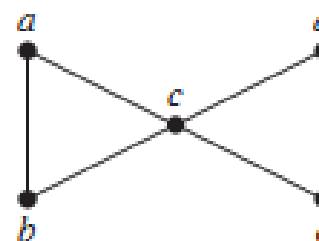
Теперь рассмотрим H .

Поскольку степени всех вершин a, b, d и e равны 2, каждое ребро, инцидентное этим вершинам, должно быть частью любого гамильтоновского цикла.

Тогда легко видеть, что в H не может существовать никакого гамильтонова цикла, поскольку любой гамильтонов цикл должен был бы содержать 4 ребра, инцидентные c , что невозможно.



G



H

Достаточные условия существования гамильтонова цикла

Хотя не известно никаких полезных **необходимых и достаточных условий** существования гамильтоновых циклов, было найдено довольно много **достаточных условий**.

Обратите внимание, что чем больше ребер в графе, тем **больше вероятность** наличия в нем гамильтонова цикла.

Более того, **добавление ребер** (но не вершин) к графу с гамильтоновым циклом приводит к созданию графа с тем же гамильтоновым циклом.

Таким образом, по мере того, как мы добавляем ребра к графу, особенно когда мы добавляем ребра к каждой вершине, мы увеличиваем вероятность того, что в этом графе существует гамильтонов цикл.

Следовательно, можно бы ожидать, что будут **достаточные условия** для существования гамильтоновых циклов, зависящие от достаточно больших **степеней вершин**.

Мы приводим здесь два наиболее важных достаточных условия.

Эти условия были обнаружены **Габриэлем А. Дираком** в 1952 году и **Эйстейном Оре** в 1960 году.

Достаточные условия существования гамильтонова цикла. Теорема Дирака.

ТЕОРЕМА 1 (ТЕОРЕМА ДИРАКА) Если G — простой граф с n вершинами, где $n \geq 3$, такой что степень каждой вершины в $G \geq n/2$, то G имеет гамильтонов цикл.

Достаточные условия существования гамильтонова цикла. Теорема Оре.

ТЕОРЕМА 2 (ТЕОРЕМА ОРЕ):

Если G — простой граф с n вершинами, где $n \geq 3$, такой, что

$\deg(u) + \deg(v) \geq n$ для каждой пары несмежных вершин u и v в G ,

тогда G имеет гамильтонов цикл .

Достаточные условия существования гамильтонова цикла. Теорема Дирака.

Теорема 1 (ТЕОРЕМА ДИРАКА) Если G — простой граф с $n \geq 3$ и $\delta \geq n/2$, то G является гамильтоновым.

Доказательство От противного. Предположим, что теорема не верна, и пусть G — максимальный негамильтонов простой граф с $n \geq 3$ и $\delta \geq n/2$. Поскольку $n \geq 3$, G не может быть полным.

Пусть u и v — несмежные вершины в G .

По выбору G , $G + \{u, v\}$ является гамильтоновым.

Более того, поскольку G негамильтонов, каждый гамильтонов цикл $G + \{u, v\}$ должен содержать ребро $\{u, v\}$.

Таким образом, существует гамильтонов путь v_1, v_2, \dots, v_n в G с началом $u = v_1$ и концом $v = v_n$

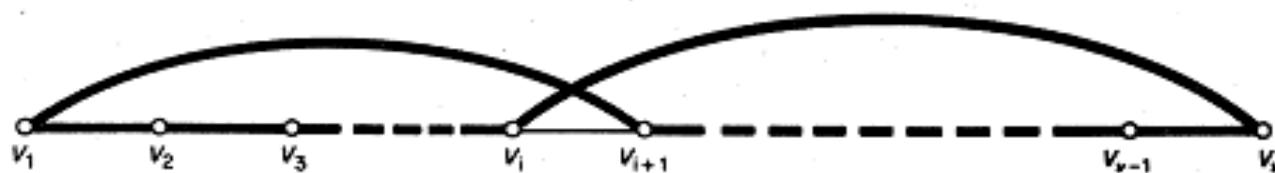
Достаточные условия существования гамильтонова цикла. Теорема Дирака.

Установим $S = \{ v_i : \{ u, v_{i+1} \} \in E \}$ и $T = \{ v_i : \{ v_i, v \} \in E \}$

Так как $v_n \notin S \cup T$, можно утверждать, что $|S \cup T| < n$ (1)

Кроме того $|S \cap T| = \emptyset$ (2)

так как если бы $S \cap T$ содержало некоторую вершину v_i , то G имел бы Гамильтонов цикл $v_1, v_2, \dots, v_i, v_n, v_{n-1}, \dots, v_{i+1}, v_1$, вопреки предположению (см. рисунок ниже).



Используя (1) и (2) получаем

$$\delta(u) + \delta(v) = |S| + |T| = |S \cup T| + |S \cap T| < n \quad (3)$$

Но это противоречит гипотезе, что $\delta(G) \geq n/2$.

Достаточные условия существования гамильтонова цикла.

И теорема Оре, и теорема Дирака дают **достаточные** условия для того, чтобы связный простой граф имел гамильтонов цикл.

Однако эти теоремы не обеспечивают **необходимых** условий существования гамильтонова цикла.

Например, граф C_5 имеет гамильтонов цикл, но не удовлетворяет условиям ни теоремы Оре, ни теоремы Дирака.

Условия существования гамильтонова цикла.

Лучшие известные алгоритмы для поиска гамильтонова цикла в графе или определения того, что такого цикла не существует имеют **в худшем случае экспоненциальную временную сложность** (по числу вершин графа).

Нахождение алгоритма, который решает эту задачу с полиномиальной временной сложностью в худшем случае, было бы крупным достижением, поскольку было показано, что эта задача является **NP-полной**.

Следовательно, существование такого алгоритма подразумевало бы, что многие другие, казалось бы, неразрешимые проблемы можно решить с помощью алгоритмов полиномиальной временной сложности в худшем случае.

Приложения гамильтоновых циклов

Гамильтоновы пути и циклы можно использовать для решения практических задач.

Например, многие приложения просят найти путь или цикл, который проходит через каждый перекресток дорог в городе, или каждый узел в сети связи только один раз.

Нахождение гамильтонова пути или цикла в соответствующей графовой модели может решить такие проблемы.

Знаменитая [задача о коммивояжере](#) требует найти кратчайший цикл, по которому коммивояжер должен посетить ряд городов. Эта задача сводится к определению гамильтонова цикла в полном графе таким образом, чтобы [общий вес](#) его ребер был как можно меньше.

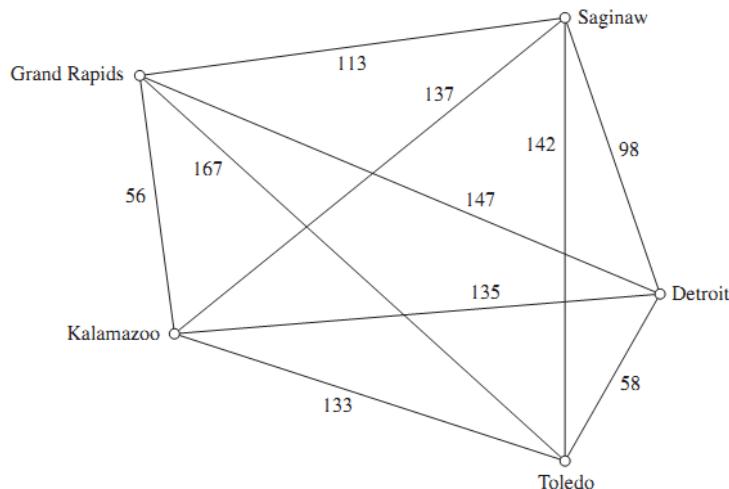
Приложения гамильтоновых циклов. Задача о коммивояжере

Рассмотрим следующую задачу:

Коммивояжер хочет посетить каждый из n городов ровно один раз и вернуться в исходную точку.

Предположим, что он хочет посетить города Детройт, Толедо, Сагино, Гранд-Рапидс и Каламазу.

В каком порядке ему следует посетить эти города, чтобы проехать минимальное общее расстояние?



Приложения гамильтоновых циклов. Задача о коммивояжере

Чтобы решить эту проблему, мы можем предположить, что продавец начинает путь в Детройте и рассмотреть все возможные способы, которыми он может посетить остальные четыре города, а затем вернуться в Детройт (начало в другом городе приведет к тем же циклам).

Всего таких циклов $4 \cdot 3 \cdot 2 \cdot 1 = 24$, но поскольку мы проходим то же самое расстояние, когда движемся по кругу в обратном порядке, нам нужно рассмотреть только 12 различных циклов, чтобы найти минимальное общее расстояние, которое ему необходимо пройти.

Мы перечислим эти 12 различных циклов и общее пройденное расстояние для каждого цикла.

Как видно из списка, минимальное общее расстояние в 458 миль преодолевается по маршруту Детройт–Толедо–Каламазу–Гранд-Рапидс–Сагино–Детройт (или в обратном направлении).

Приложения гамильтоновых циклов. Задача о коммивояжере

<i>Route</i>	<i>Total Distance (miles)</i>
Detroit–Toledo–Grand Rapids–Saginaw–Kalamazoo–Detroit	610
Detroit–Toledo–Grand Rapids–Kalamazoo–Saginaw–Detroit	516
Detroit–Toledo–Kalamazoo–Saginaw–Grand Rapids–Detroit	588
Detroit–Toledo–Kalamazoo–Grand Rapids–Saginaw–Detroit	458
Detroit–Toledo–Saginaw–Kalamazoo–Grand Rapids–Detroit	540
Detroit–Toledo–Saginaw–Grand Rapids–Kalamazoo–Detroit	504
Detroit–Saginaw–Toledo–Grand Rapids–Kalamazoo–Detroit	598
Detroit–Saginaw–Toledo–Kalamazoo–Grand Rapids–Detroit	576
Detroit–Saginaw–Kalamazoo–Toledo–Grand Rapids–Detroit	682
Detroit–Saginaw–Grand Rapids–Toledo–Kalamazoo–Detroit	646
Detroit–Grand Rapids–Saginaw–Toledo–Kalamazoo–Detroit	670
Detroit–Grand Rapids–Toledo–Saginaw–Kalamazoo–Detroit	728

Приложения гамильтоновых циклов. Задача о коммивояжере

Задача о коммивояжере заключается в поиске цикла **минимального общего веса** во взвешенном, полном, неориентированном графе, который посещает каждую вершину ровно один раз и возвращается в исходную точку.

Это эквивалентно поиску **гамильтонова цикла** с минимальным общим весом в **полном графе**, поскольку каждая вершина в цикле посещается ровно один раз.

Самый простой способ решения задачи о коммивояжере — это рассмотреть все возможные **гамильтоновы циклы** и выбрать один с **минимальной общей длиной**.

Сколько циклов нам нужно рассмотреть, чтобы решить задачу, если в графе n вершин?

Приложения гамильтоновых циклов. Задача о коммивояжере

После выбора начальной точки необходимо исследовать $(n - 1)!$ различных гамильтоновых циклов, поскольку для второй вершины существует $n - 1$ вариантов, для третьей вершины — $n - 2$ варианта и т. д. Поскольку гамильтонов цикл можно пройти в обратном порядке, нам нужно рассмотреть только $(n - 1)!/2$ циклов, чтобы найти ответ.

Заметим, что $(n - 1)!/2$ растет чрезвычайно быстро.

Попытка решить задачу коммивояжера таким способом, когда имеется всего несколько десятков вершин, нецелесообразна.

Например, при 25 вершинах необходимо будет рассмотреть в общей сложности $24!/2$ (приблизительно $3,1 \times 10^{23}$) различных гамильтоновых циклов.

Если бы для проверки каждого гамильтонова цикла требовалась всего 1 наносекунда (10^{-9} секунды), то для нахождения гамильтонова цикла минимальной длины в этом графе с помощью методов полного перебора потребовалось бы в общей сложности около 10 миллионов лет .

Приложения гамильтоновых циклов. Задача о коммивояжере

Поскольку задача коммивояжера имеет как практическое, так и теоретическое значение, немало усилий было направлено на разработку эффективных алгоритмов ее решения.

Однако **неизвестен алгоритм с полиномиальной временной сложностью в худшем случае для решения этой задачи.**

Более того, если бы был открыт полиномиальный алгоритм наихудшей временной сложности для задачи коммивояжера, многие другие сложные задачи также могли бы быть решены с использованием полиномиальных алгоритмов наихудшей временной сложности (например, определение того, является ли предложение с n переменными тавтологией).

Это следует из теории NP -полноты.

Приложения гамильтоновых циклов. Задача о коммивояжере

Практическим подходом к задаче о коммивояжере, когда необходимо посетить много вершин, является использование [приближенных алгоритмов](#).

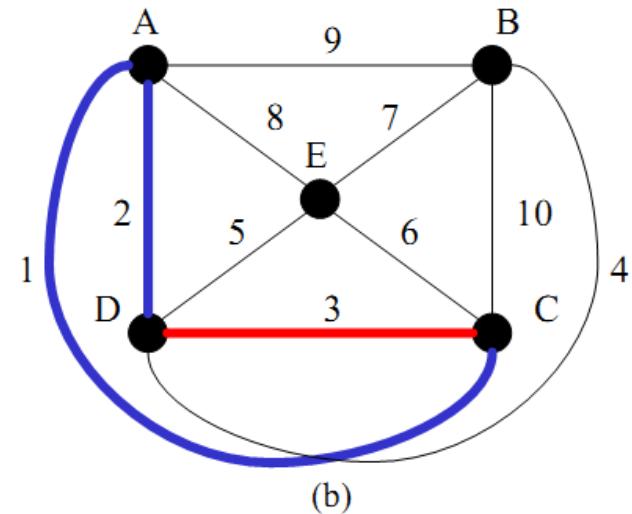
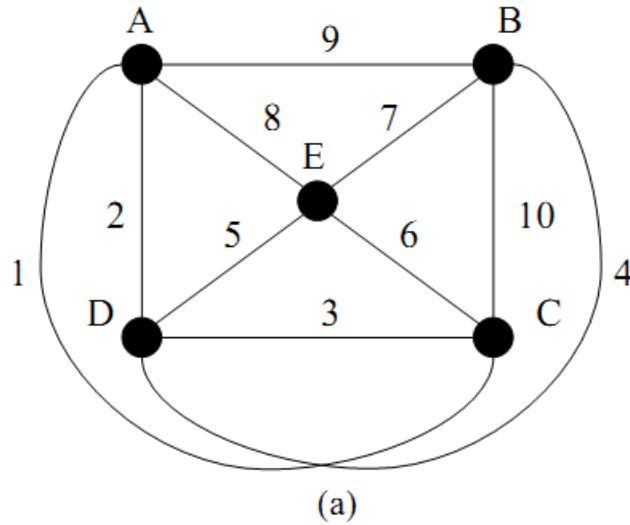
Это алгоритмы, которые не обязательно дают точное решение задачи, но вместо этого гарантированно дают решение, близкое к точному.

На практике были разработаны алгоритмы, которые позволяют решать задачи о коммивояжере с числом вершин до 1000 с точностью до 2%, используя всего несколько минут машинного времени.

Алгоритм самой дешевой связи (Cheapest Link Algorithm, CLA)

1. Выбрать ребро с наименьшим весом (самое «дешевое» ребро), случайным образом выбирая один из равнозначных вариантов.
 2. Продолжать выбирать «самое дешевое» ребро, если только оно
 - (а) не замыкает **меньший цикл** ИЛИ
 - (б) не приводит к тому, что из одной вершины выходят 3 выбранных ребра
- Продолжать, пока не завершится гамильтонов цикл.

Алгоритм самой дешевой связи (Cheapest Link Algorithm, CLA)

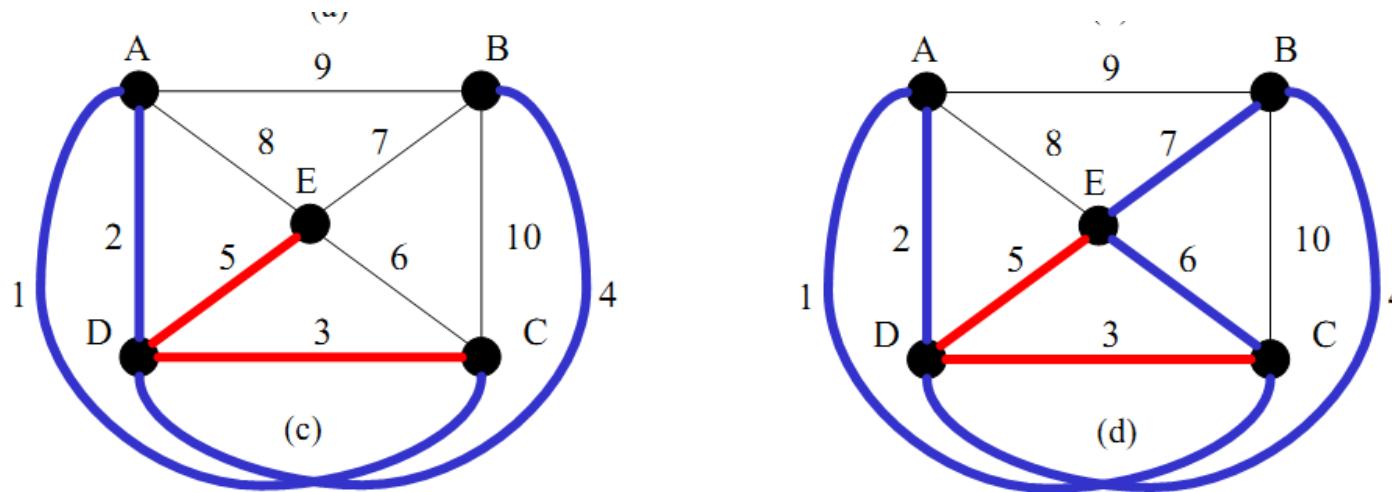


Пример:

Применим этот алгоритм к графу K_5 :

- 1) Выбираем ребро **AC (вес 1)** (минимальный вес).
- 2) Затем, поскольку никакие ограничения не будут нарушены, мы добавляем ребро **AD (вес 2)**.
- 3) Теперь, если мы попытаемся добавить ребро **DC (вес 3)**, мы получим цикл **ACDA**, поэтому мы его не добавляем.

Алгоритм самой дешевой связи (Cheapest Link Algorithm, CLA)



Вместо этого мы берем следующее ребро с наименьшим весом DB (вес 4) и добавляем его.

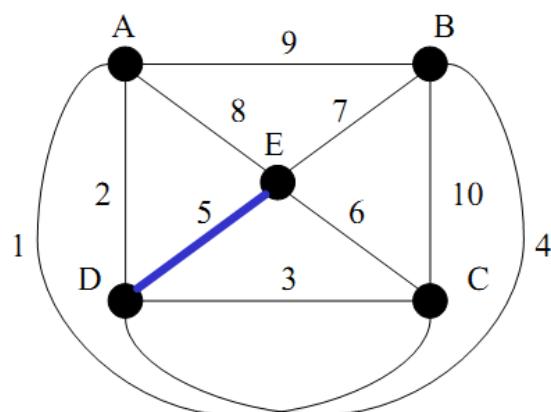
Если мы добавим ребро ED (вес 5), то вершина D будет иметь степень 3, поэтому мы его не используем.

Вместо этого мы последовательно добавляем EC (вес 6) и EB (вес 7), так что мы получаем гамильтонов цикл ADBECA с общей стоимостью 20

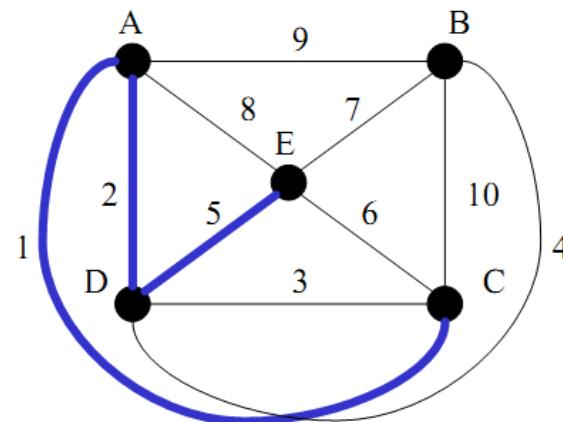
Алгоритм ближайшего соседа (NNA)

1. Начать с некоторой вершины (можно считать, что это родной город)
2. Перейти в вершину, в которой **вы еще не были**, путь в которую имеет наименьший вес. Если есть несколько эквивалентных вариантов, выбрать любой случайным образом.
3. Продолжать, пока не пройдете все вершины.
4. Вернуться в исходную вершину.

Алгоритм ближайшего соседа (NNA)



(a)



(b)

Пример Применить NNA к графу K_5

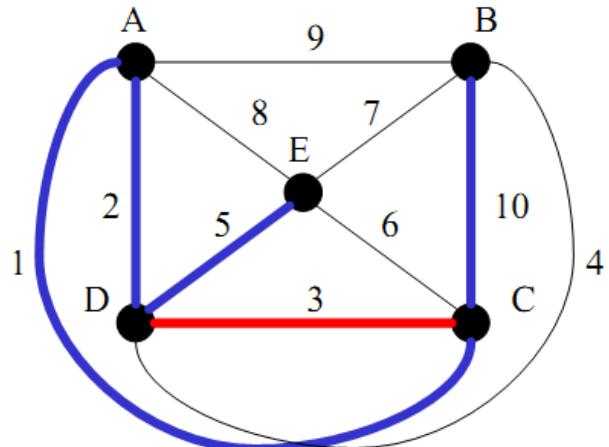
Выбираем случайным образом вершину **E** в качестве начальной.

Выбираем ребро с наименьшей стоимостью, инцидентное **E**, **ED** (вес 5) .

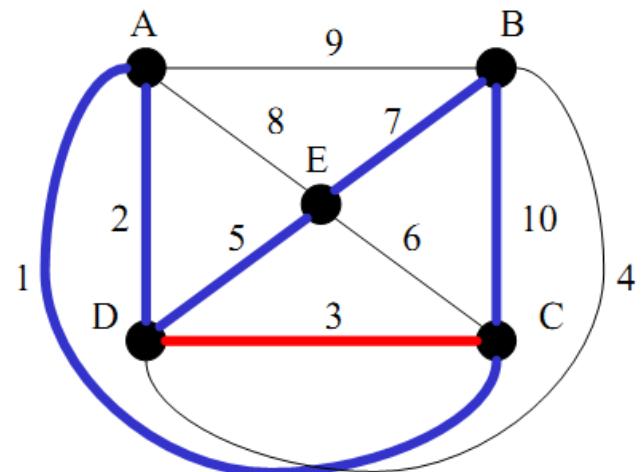
Таким же образом добавляем **DA**(вес 2) и **AC**(вес 1).

Здесь мы **не можем использовать** ребро наименьшей стоимости **CD**(вес 3), так как вершина **D** уже посещена

Алгоритм ближайшего соседа (ННА)



(c)



(d)

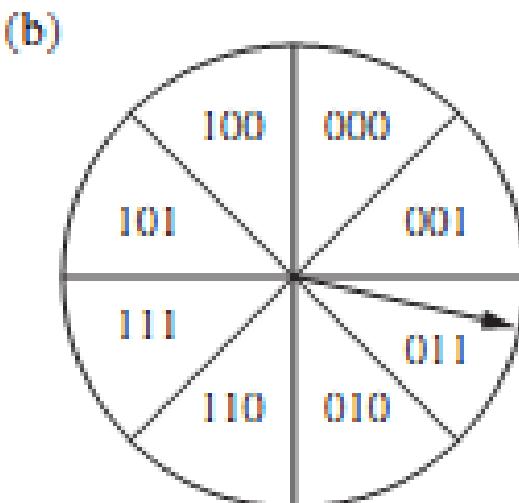
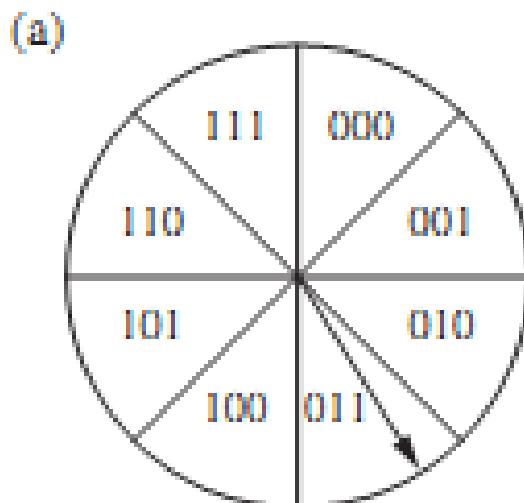
Поэтому перемещаемся в В через ребро СВ(вес 10) и, наконец, получаем цикл EDACBE общей стоимостью 25 (хуже предыдущего)

Приложения гамильтоновых циклов. Коды Грея

Теперь опишем менее очевидное применение гамильтоновых циклов к [кодированию](#).

Положение вращающейся стрелки можно представить в цифровом виде. Один из способов сделать это — разделить круг на 2^n дуг одинаковой длины и сопоставить каждой дуге битовую строку длины n .

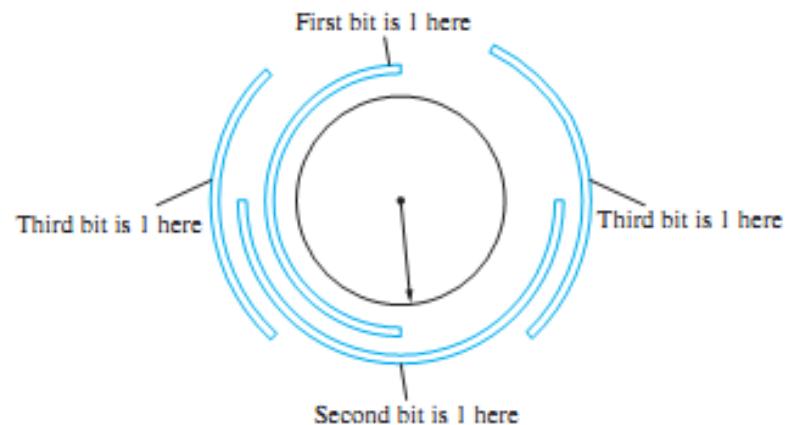
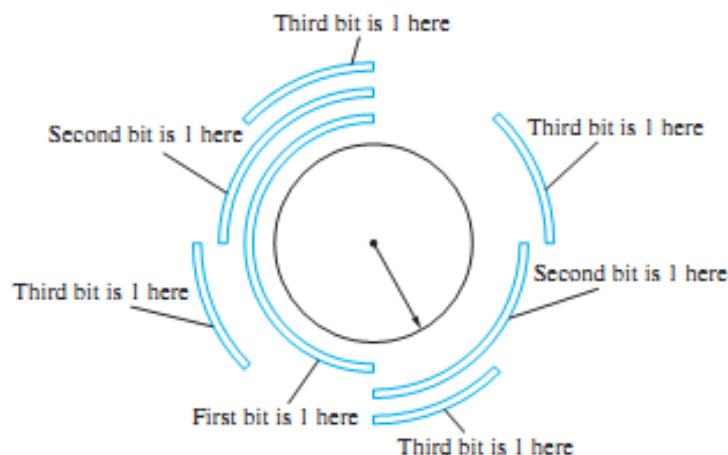
Ниже показаны два способа сделать это с использованием битовых строк длины 3.



Приложения гамильтоновых циклов. Коды Грея

Цифровое представление положения указателя можно определить с помощью набора [из \$p\$ контактов](#).

Каждый контакт используется для считывания [1 бита](#) в цифровом представлении положения.



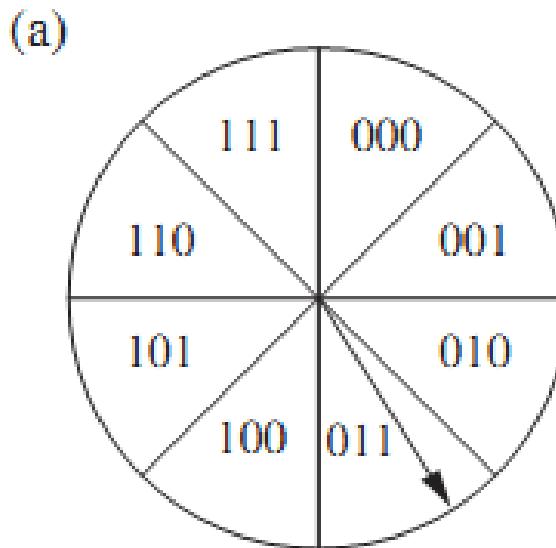
Приложения гамильтоновых циклов. Коды Грея

Когда указатель находится вблизи границы двух дуг, может быть допущена ошибка в определении его положения.

Это может привести к серьезной ошибке при чтении битовой строки.

Например, в приведенной ниже схеме кодирования, если при определении положения указателя допущена небольшая ошибка, вместо 011 будет прочитана битовая строка **100**.

Все три бита неверны!



Приложения гамильтоновых циклов. Коды Грея

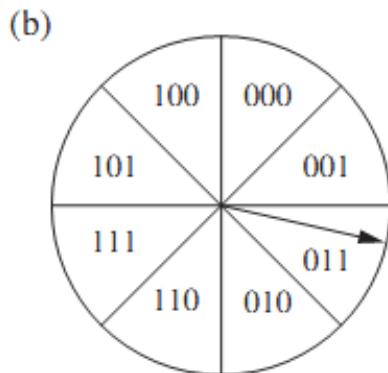
Чтобы минимизировать влияние ошибки при определении положения указателя,

При присваивании битовых строк 2^n дугам

- необходимо сделать так, чтобы битовые строки, соответствующие соседним дугам, отличались только на **1 бит**.

Именно такая ситуация наблюдается в схеме кодирования ниже.

- В этом случае ошибка в определении положения указателя дает битовую строку **010** вместо **011**. Ошибка только в **одном бите**.



Распределение битовых строк представляет собой **код Грея**.

Приложения гамильтоновых циклов. Коды Грея

Код Грея — это маркировка дуг окружности таким образом, что соседние дуги маркируются битовыми строками, отличающимися ровно на 1 бит.

Мы можем найти код Грея, перечислив все строки битов длины n таким образом, что каждая строка отличается от предыдущей строки битов ровно на 1 позицию, а последняя строка отличается от первой ровно на одну позицию.

Мы можем смоделировать эту проблему, используя n -куб Q_n .

- Для решения этой задачи необходимо найти гамильтонов цикл в Q_n .

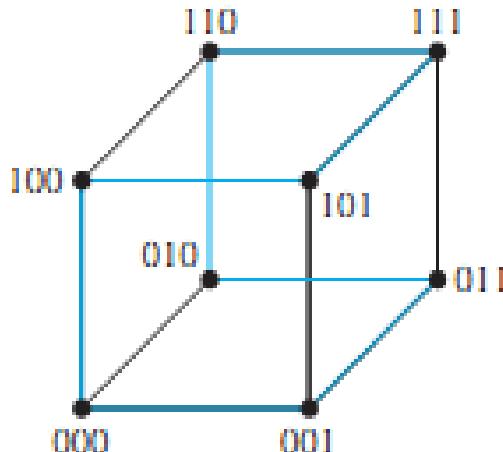
Приложения гамильтоновых циклов. Коды Грея

Такие гамильтоновы циклы легко найти.

Например, гамильтонов цикл для Q_3 показан ниже.

Последовательность битовых строк, отличающихся ровно на один бит, создаваемая этим гамильтоновым циклом, равна **000, 001, 011, 010, 110, 111, 101, 100**.

Коды Грея названы в честь Фрэнка Грея, который изобрел их в 1940-х годах в лабораториях AT&T Bell для минимизации влияния ошибок при передаче цифровых сигналов.



Гамильтонов цикл для Q_3 .

- Ваши вопросы?
- Контакты лектора:
aapanovich_09@mail.ru