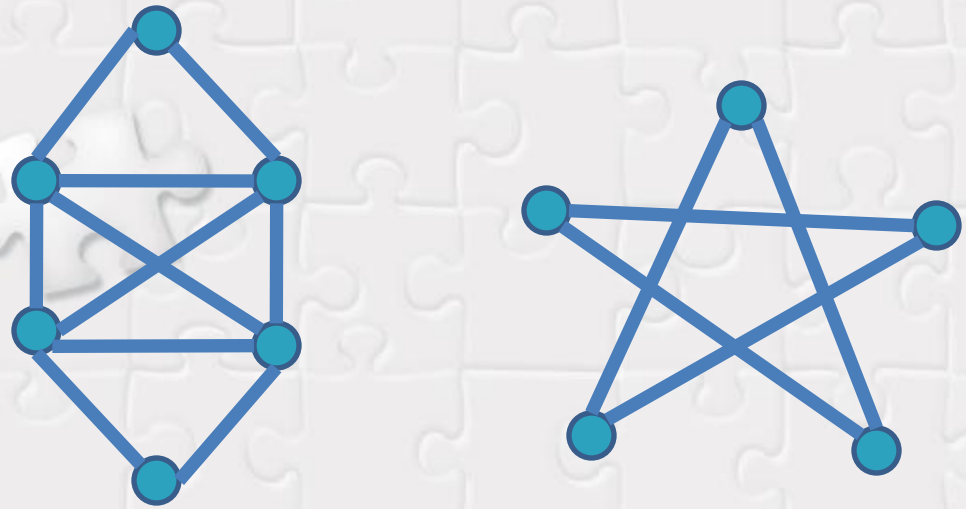
Лекция 8
ЭЙЛЕРОВЫ ГРАФЫ

Эйлеровы графы

Если граф имеет цикл (не обязательно простой), содержащий все ребра графа по одному разу, то такой цикл называется **эйлеровым циклом**, а граф называется **эйлеровым графом**.

Эйлеров цикл содержит не только все ребра, но и все вершины графа (возможно по несколько раз). Ясно, что эйлеровым может быть только связный граф.

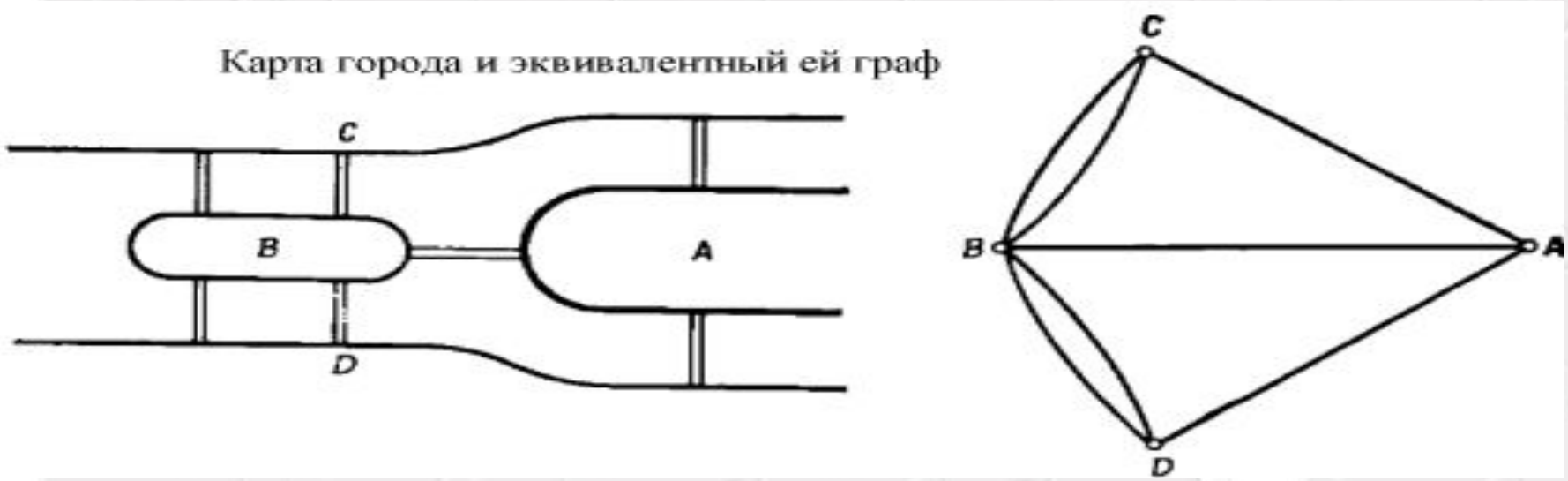
Эйлеров граф можно нарисовать на бумаге, не отрывая от нее карандаша. Вышеопределенные понятия распространяются аналогично на мультиграфы.



Леонард Эйлер **(1707-1783) первым в своей знаменитой задаче о Кёнигсбергских мостах рассмотрел вопрос о существовании таких циклов в графах.**

Кёнигсберг (Калининград) расположен на обоих берегах реки Преголя и на двух островах этой реки. Берега реки и два острова соединены семью мостами, как показано на карте.

1736 г.: Можно ли начав с некоторой точки, совершить прогулку и вернуться в исходную точку, пройдя по каждому мосту ровно один раз?



Теорема

Если неориентированный граф G связен и в нем более одной вершины, то следующие утверждения эквивалентны:

1) G — эйлеров граф.

2) Каждая вершина G имеет четную степень.

3) Множество ребер G можно разбить на простые циклы.

1) → 2): G – эйлеров граф → эйлеров цикл проходя через каждую вершину, входит в нее по одному ребру, выходит по другому → каждая вершина инцидентна четному числу ребер. Цикл содержит все ребра → четность степей всех вершин

2) → 1): все степени четны → построим цикл, содержащий все ребра. Начнем строить цепь P_1 из произвольной вершины v_1 и будем продолжать ее насколько это возможно, выбирая каждый раз новое ребро. Т.к. Степени вершин четны, то

попав в очередную отличную от v_1 вершину, будем иметь в распоряжении еще непройденное ребро. Добавим его в цепь P_1. Построение цепи закончится в вершине v_1. Если P_1 содержит все ребра, то построение закончено. Иначе удалим из графа все ребра, принадлежие P_1.

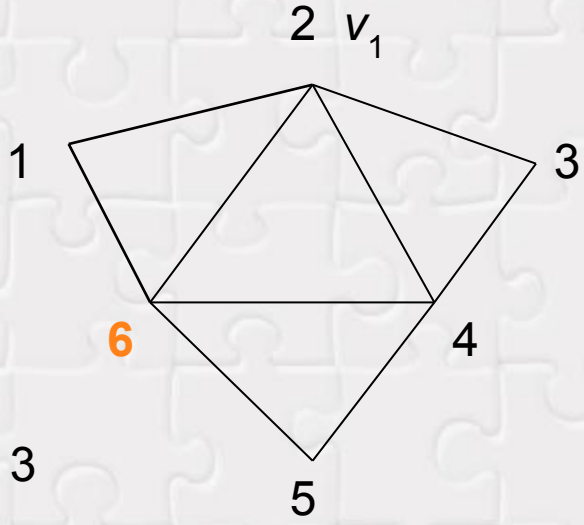
Рассмотрим граф G_1, полученный в результате удаления этих ребер. G и P_1 имели вершины только четных степей $\rightarrow$ G_1 будет иметь вершины только четных степей. В силу связности P_1 и G_1 будут иметь хотя бы одну общую вершину v_2.

Начиная с вершины v_2 построим цикл P_2 в G_1. Пусть P_1' и P_1'' – части цикла P_1 от v_1 до v_2 и от v_2 до v_1, соответственно. Получим новый цикл: $P_3 = P_1' \cup P_2 \cup P_1''$, который начинается в v_1, проходит по всем ребрам P_1' до v_2, затем все ребра P_2 и возвращается в v_1 по ребрам из P_1''.

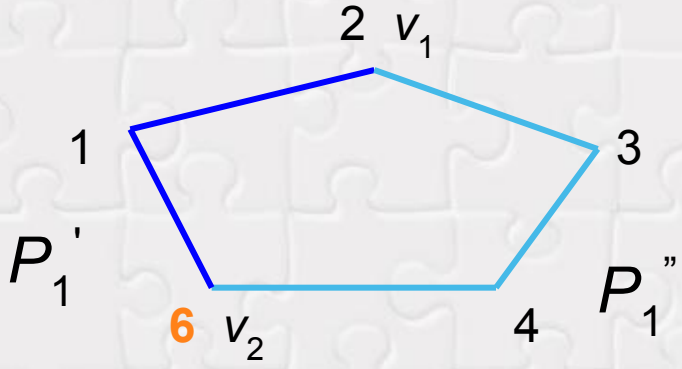
Аналогично продолжим процесс до получения эйлерова цикла.

Пример

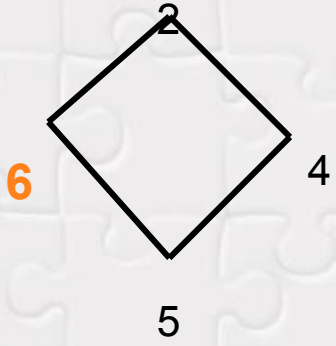
Граф G



Цепь P_1



Цепь P_2



Алгоритмы поиска эйлерова цикла

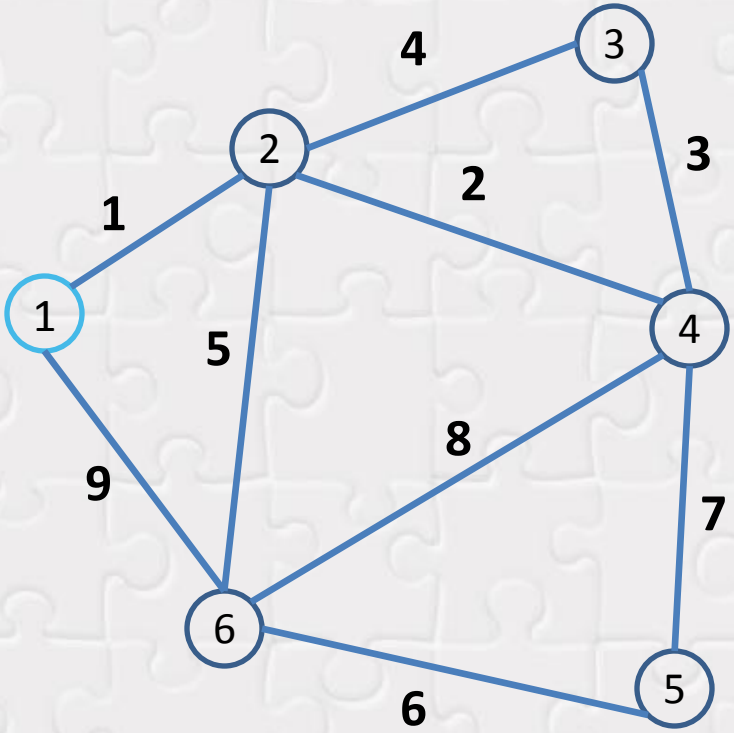
Алгоритм Флёри (сложность O(n*(n+m))

Требуется занумеровать ребра графа числами 1, 2, …, |*E*|, так, чтобы номер, присвоенный ребру, указывал, каким по счету это ребро проходится в эйлеровом цикле.

1. Начиная с произвольной вершины *u*, присваиваем произвольному ребру (*u*,*v*) номер 1. Затем вычеркиваем ребро (*u*,*v*) из графа и переходим в вершину *v*.
2. Пусть *w* – вершина, в которую мы пришли в результате выполнения предыдущего шага, и *k* – номер, присвоенный некоторому ребру на этом шаге. Выбираем любое ребро, инцидентное вершине *w*, причем *мост* выбираем только в том случае, если нет других возможностей; присваиваем выбранному ребру номер *k* + 1, проходим по нему в следующую вершину и вычеркиваем его .
3. Алгоритм заканчивается, когда все ребра вычеркнуты

Мост – ребро, удаление которого из графа приводит к тому, что граф распадается на несколько компонент связности.

Пример



Алгоритм поиска эйлерового цикла (O(*n*+*m*))

Начиная с произвольной вершины, строим путь, удаляя ребра и запоминая вершины в стеке, до тех пор пока множество смежности очередной вершины не окажется пустым, что означает, что путь удлинить нельзя.

Заметим, что при этом мы с необходимостью придем в ту вершину, с которой начали. В противном случае это означало бы, что вершина *v* имеет нечетную степень, что невозможно по условию.

Таким образом, из графа были удалены ребра цикла, а вершины цикла были сохранены в стеке *S*.

Заметим, что при этом степени всех вершин остались четными. Далее вершина *v* выводится в качестве первой вершины эйлерового цикла, а процесс продолжается, начиная с вершины, стоящей на вершине стека.

Алгоритм

Вход: эйлеров граф $G(V, E)$, заданный списками смежности ($\Gamma[v]$ — список вершин смежных с вершиной v).

Выход: последовательность вершин эйлерового цикла.

```
S := Ø { стек для хранения вершин }
выбрать v ∈ V { произвольная вершина }
v → S { положить v в стек S }
пока S ≠ Ø выполнять
 v ← S;  v → S { v — верхний элемент стека }
 если Γ[v] = Ø
 то  v ← S; вывод v
 иначе
        выбрать u ∈ Γ[v]
        { взять первую вершину из списка смежности }
 u → S
 Γ[v] = Γ[v] \ {u};
        Γ[u] = Γ[u] \ {v} { удалить ребро (v, u) }
 конец если
конец пока
```

Рекурсивная реализация

```
void cycle_search(u) {
    for (берем любое непройденное ребро (u,v)) {
      (u,v) – отметить и удалить из списка;
     cycle_search(v);
    }
    вывести_вершину (u);
}
```

Теорема

Граф G 2-раскрашиваемый $\Leftrightarrow$ G – эйлеров

Задача о рыбе

Дан набор костяшек домино.

Нужно определить, можно ли из них построить рыбу (так расположить костяшки, чтобы они были все использованы, и последовательность начиналась и заканчивалась одним и тем же числом).