Эйлеровы графы

Eulerian graph

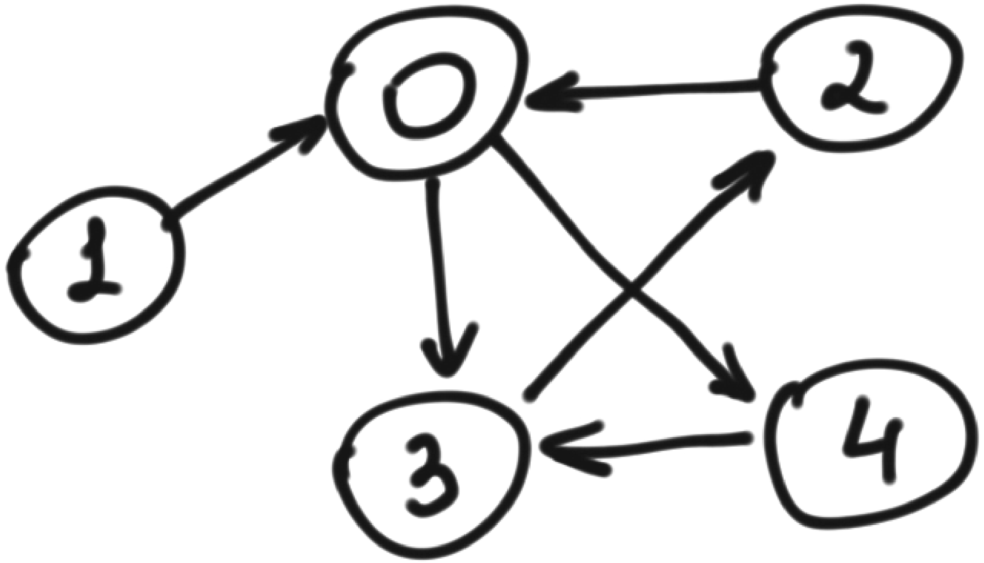
Эйлеровость

Эйлеров путь - путь в графе, проходящий по каждому ребру графа ровно 1 раз.

Полуэйлеров граф - граф, в котором есть эйлеров путь.

Эйлеров цикл - замкнутый эйлеров путь.

Эйлеров граф - граф, в котором есть эйлеров цикл.



Критерии эйлеровости

Теорема 1.1 (критерий эйлеровости для неориентированных графов)

Граф G *- эйлеров* $\Leftrightarrow G$ *содержит не более одной компоненты связности с ребрами и* $\forall v \in G.V\ : deg(v) - \text{четна}$

Критерии эйлеровости

Теорема 1.1 (критерий эйлеровости для неориентированных графов)

Граф G *- эйлеров* $\Leftrightarrow G$ *содержит не более одной компоненты связности с ребрами и* $\forall v \in G.V : deg(v) - \text{четна}$

Доказательство.
$\Rightarrow$ Рассмотрим эйлеров цикл и пройдем по нему. Так как путь замкнут, в одну и ту же вершину зайдем и выйдем одинаковое количество раз.

$\Leftarrow$ Докажем конструктивно, предъявив алгоритм. $\blacksquare$

Критерии эйлеровости

Теорема 1.2 (критерий полуэйлеровости для неориентированных графов)

Граф G *- полуэйлеров* $\Leftrightarrow G$ *содержит не более одной компоненты связности с ребрами и* $\forall v \in G.V : deg(v) - \text{четна}$*, кроме, может быть, 2-х.*

Доказательство.

Если у каждой вершины степень четна, то сводится к **Теореме 1.1.**

Если есть 2 вершины с нечетной степенью, то введем фиктивное ребро между этими вершинами и сведем к **Теореме 1.1.** ■

Критерии эйлеровости

Теорема 2.1 (критерий эйлеровости для ориентированных графов)

Орграф G - эйлеров $\Leftrightarrow G$ содержит не более одной компоненты слабой связности с ребрами и $\forall v \in G.V : deg_+(v) = deg_-(v)$

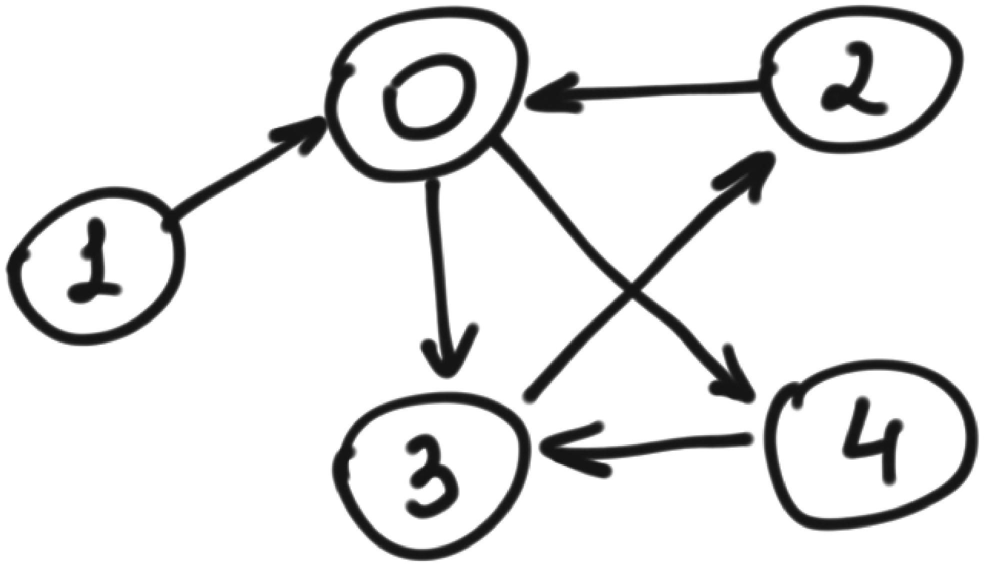
Теорема 2.2 (критерий полуэйлеровости для ориентированных графов)

Орграф G - полуэйлеров $\Leftrightarrow G$ содержит не более одной компоненты слабой связности с ребрами и $\forall v \in G.V : deg_+(v) = deg_-(v)$, кроме, может быть, 2-х, у одной из которых полустепень захода на 1 меньше полустепени исхода, а у второй - наоборот.

Доказательство. Аналогично **Теоремам 1.1. и 1.2** ■

Алгоритм поиска эйлерова цикла: идея

1. Будем искать цикл рекурсивно с помощью *DFS*
2. Но *DFS* посещает вершины по 1 разу, в то время как эйлеров цикл может заходить в вершины несколько раз
3. Поэтому помечать посещенными будем не вершины, а ребра.
4. По завершении обработки вершины добавляем ее в начало массива пути.



Алгоритм поиска эйлерова цикла

```
def DfsEulerCycle(G, p, v):  # G - граф, v - вершина, p - родитель v
  for u in G.neighbors(v):
    G.neighbors(v).erase(u)  # помечаем ребро посещенным
    DfsEulerCycle(G, v, u)
  euler_cycle.push_front((p, v))  # добавляем ребро в путь
```

Сложность $O(E+V)$ при использовании списков смежности.

Отметим, что для корректной работы алгоритма должны быть выполнены соответствующие необходимые условия из критерия эйлеровости.
Их нужно проверять отдельно.

Корректность

Рассмотрим случай неориентированного графа. Остальные - аналогично.

Утверждение 1. *Каждое ребро в массиве euler_cycle содержится ровно один раз.*

Доказательство. Каждое ребро посетим ровно 1 раз и положим его в массив *euler_cycle*. ■

Утверждение 2. *В момент выписывания ребра степени всех вершин четны.*

Корректность

Рассмотрим случай неориентированного графа. Остальные - аналогично.

Утверждение 1. *Каждое ребро в массиве euler_cycle содержится ровно один раз.*

Доказательство. Каждое ребро посетим ровно 1 раз и положим его в массив *euler_cycle*. ■

Утверждение 2. *В момент выписывания ребра степени всех вершин четны.*

Доказательство. Изначально степени всех вершин четны. Допустим, запустили поиск цикла из стартовой вершины s. Тогда алгоритм не может впервые выписать ребро с концом в вершине $u \neq s$, так как изначально степень каждой вершины четна и из нее всегда есть куда пойти. Значит рекурсивный вызов приведет снова в s, у которой не останется ребер. Поэтому будет выписано ребро $(p(s), s)$. Затем откатимся в $p(s)$, и продолжим обход на графе с четными степенями (то есть свели к предыдущему случаю). ■

Корректность

Утверждение 3. *Последовательность выписанных ребер образует путь.*

Корректность

Утверждение 3. *Последовательность выписанных ребер образует путь.*

Доказательство.

Пусть выписали ребро (u, v). В этот момент степени всех вершин четны и алгоритм продолжает обход из вершины u. Следовательно, следующий момент, когда станет некуда идти (то есть придется выписывать ребро), настанет в вершине u. То есть следующим будет выписано ребро $(p(u), u)$ ■

Теорема (о корректности поиска эйлерова цикла).

Алгоритм корректно находит эйлеров цикл.

Доказательство. Следует из **Утверждений 1 и 3** ■

Удивительное рядом

Существует аналогичная задача для вершин:

Гамильтонов цикл - цикл, проходящий по каждой вершине графа ровно 1 раз.

Задача поиска гамильтонова цикла является NP-полной. То есть, скорее всего, для нее не существует эффективного решения.

Но вы все равно попытайтесь за доп. балл к домашке.