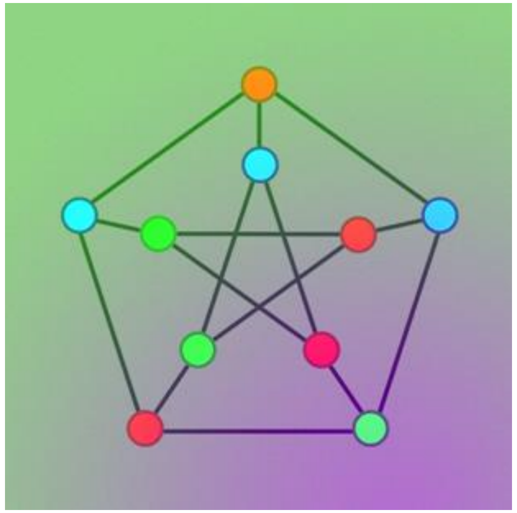
Хроматическое число графов

Хроматическое число графа

- Раскраска вершин графа в разные цвета называется правильной, если смежные вершины раскрашены в разные цвета.
- Хроматическим числом графа называется минимальное число цветов, которыми можно правильно раскрасить вершины графа.



- Граф Петерсена

Раскраска вершин графа

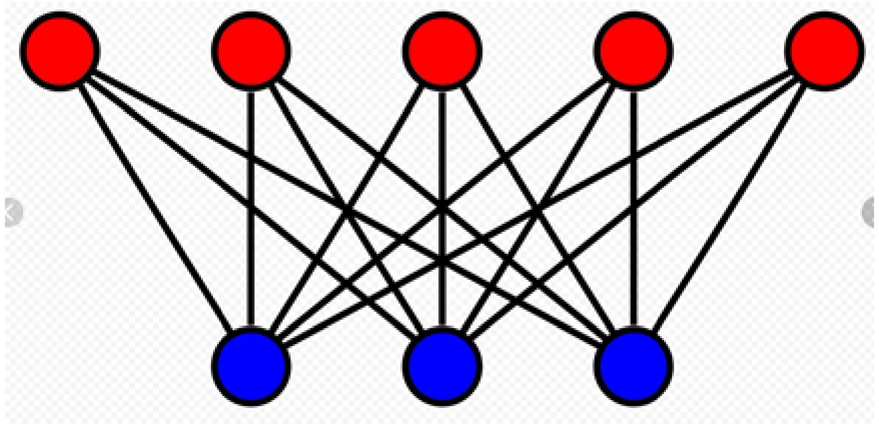
- Пусть k — натуральное число. Раскраской графа G = <V, E> в k цветов, или просто k -раскраской, называется отображение f из множества V в множество {1, 2, . . . , k}.
- Если при этом f (v) = i для некоторой вершины v ∈ V, то будем говорить, что вершина v раскрашена в i -й цвет.
- Раскраска f графа называется правильной, если f (u) != f (v) для любых двух смежных вершин u и v этого графа.
- Если существует правильная k-раскраска графа G, то G называют k-раскрашиваемым.

Хроматическое число графа

- Число k называется хроматическим числом графа G и обозначается через $\chi(G)$, если существует правильная k-раскраска графа G, но не существует его правильной $(k - 1)$-раскраски.
- Если рассматривать полный граф, содержащий к-вершин, то, очевидно, что его хроматическое число будет равно к.

Графы с малым хроматическим числом

- Пусть G — обыкновенный граф. Тогда:
- 1) $\chi(G) = 1$ тогда и только тогда, когда G — пустой граф.
- 2) $\chi(G) = 2$ тогда и только тогда, когда G — непустой двудольный граф.



Если наибольшая из степеней вершин графа G равна u , то этот граф u+1 - раскрашиваем.

- **Доказательство** Проведем индукцию *по* числу вершин в G.
- Пусть G — *граф* с n вершинами; если из него удалить произвольную вершину v вместе с *инцидентными* ей ребрами, то в оставшемся графе будет n-1 вершин, причем степени вершин *по*-прежнему не превосходят u.
- *По* предположению индукции этот *граф* u+1 -раскрашиваем.
- Из этой раскраски получится u+1 -раскладка для G, если окрасить вершину v цветом, отличным от тех, которыми окрашены смежные с ней вершины, а их не более чем u.

Теорема (Брукса). Пусть G — связный граф, не являющийся полным; если наибольшая из степеней его вершин равна u(u>2), то он u - раскрашиваем.

- Доказательство так же проводится индукцией по числу вершин.

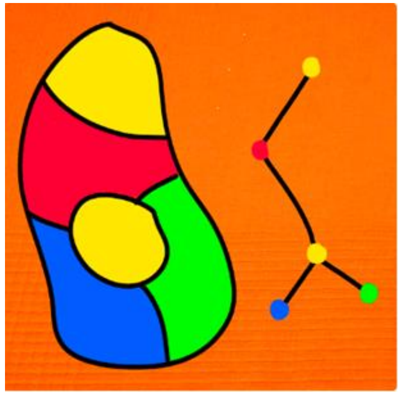
Задача о четырех красках

- Представьте себе географическую карту поверхности планеты (шара), на которой есть только суша; каждая точка поверхности принадлежит какой-то стране. Все страны односвязные — представляют собой единый кусок без дырок.

- Картографу нужно раскрасить карту так, чтобы никакие две соседние страны не были одного цвета. Соседние страны — это такие, у которых есть общая граница ненулевой длины;
- Как много красок нужно картографу, чтобы раскрасить карту по таким правилам?
- Оказывается, четыре, — в этом и состоит утверждение теоремы о четырех красках.

Перевод задачи на язык графов

- Надо сжать каждую страну до одной цветной точки, и соединить точки отрезками, если у этих стран есть общая граница. Получится граф, у которого ребра не пересекаются (планарный).



Вершины любого планарного графа можно раскрасить 4 или меньшим числом красок.

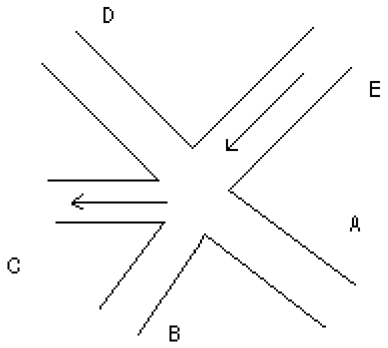
История доказательства теоремы о 4 красок

- Но ее долго не могли доказать, и она оставалась известной как задача о четырех красках. Считается, что в середине XIX века ее поставил Фрэнсис Гутри в письме своему брату.
- И только в 1974 году Аппель и Хакен с применением компьютера показали, что четырех красок достаточно, и это вызвало серьезные разногласия в математическом сообществе.
- Доказательство Аппеля и Хакена было первым компьютерным доказательством в математике. И до сих пор такой подход остается неоднозначным; общепринятым его тоже нельзя назвать.

Задача, которая сводится к задаче раскраски вершин графа

- Задача определить режимов работы светофоров на сложных перекрестках.
- Задача транспортная, но на первый взгляд решение не предполагает использование теории графов.
- Вершины графа – это допустимые на перекрестке повороты, причем проезд по прямой считается поворотом на 0 градусов. Вершины соединяются ребром, если повороты конфликтуют. Задача разбить это множество на группы так, чтобы повороты в одной группе могли выполнятся одновременно.
- Можно каждой группе поставить в соответствие режим работы светофора на перекрестке.
- Желательно минимизировать количество режимов работы светофора.

Пример. Постановка задачи.



Дороги С, Е односторонние, все остальные дороги двухсторонние.

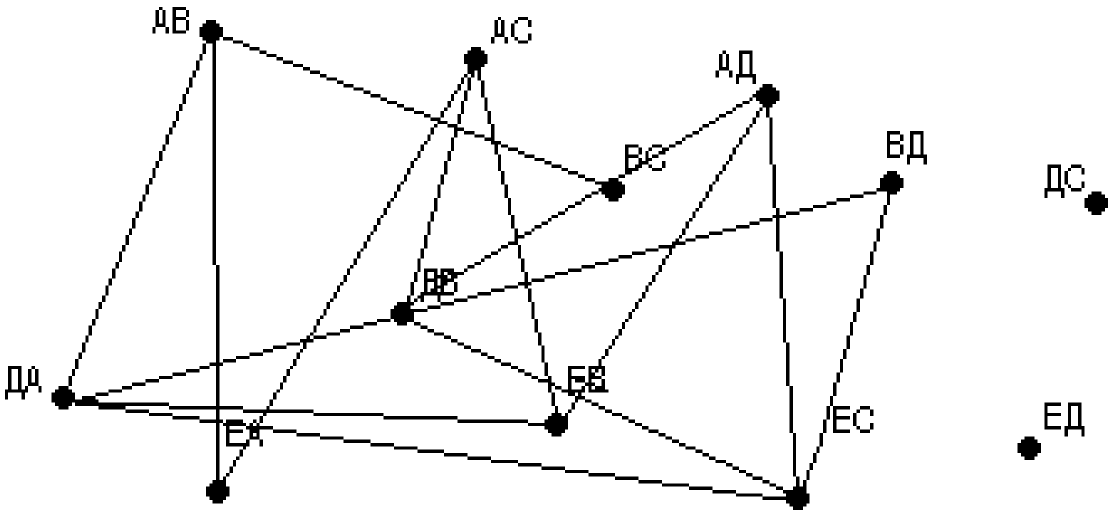
Всего возможно 13 поворотов.

Некоторые повороты, например, АВ и ЕД могут выполняться одновременно,

другие, например, АД и ЕС пересекаются,

потому одновременно их выполнять нельзя.

Пример. Для решения задачи построим граф



В рамках этой модели можно использовать решение, которое дает задача раскраски графов.

Проблемы возникающие при решении задачи раскраски графов.

- Задача раскраски произвольного графа минимальным количеством цветов принадлежит классу NP- полных задач.
- Для раскраски графа это означает, что сначала для закраски вершины используется 1 цвет затем 2, 3 и т д., При этом, передираем все варианты до тех пор, пока не будет получена правильная раскраска.
- Оптимальные решение этой задачи требует больших вычислительных затрат.

Варианты решения проблемы

- 1) Если граф не большой, то можно пытаться найти решение, перебрав все варианты.
- 2) Найти дополнительную информацию о исходной задачи, которая позволит не перебирать лишние варианты.
- 3) Изменить постановку задачи. Искать не оптимальное решение, а приемлемое, близкое к оптимальному.

Жадный алгоритм

- 1шаг. Выбираем произвольную, незакрашенную вершину и назначаем ей новый цвет.
- 2 шаг. Просматриваем список незакрашенных вершин и для каждой из них определяем, соединяется ли она ребром с вершиной, уже закрашенной в новый цвет. Если не соединяется, то к этой вершине применяем новый цвет. Просматриваем последовательно все вершины.
- 3 шаг. Проверяем, остались ли незакрашенные вершины, если – да, то переходим к шагу 1, в противном случае завершаем работу.

Этот алгоритм называют «жадным» из- за того, что каждый цвет применяется к максимально большому числу вершин без пропуска их или перекраске ранее закрашенных.

Пример получения не оптимального решения.